5-2016

Analyzing the phenologic dynamics of kudzu (Pueraria montana) infestations using remote sensing and the normalized difference vegetation index.

Faye Peters
University of Louisville

ANALYZING THE PHENOLOGIC DYNAMICS OF KUDZU (*PUERARIA MONTANA)* INFESTATIONS USING REMOTE SENSING AND THE NORMALIZED DIFFERENCE VEGETATION INDEX

By

Faye Peters
B.A., University of New Mexico, 2007

A Thesis
Submitted to the Faculty of the
College of Arts and Sciences of the University of Louisville
In Partial Fulfillment of the Requirements
For the Degree of

Master of Science
in Applied Geography

Department of Geography and Geosciences
University of Louisville
Louisville, KY

May 2016

ANALYZING THE PHENOLOGIC DYNAMICS OF
KUDZU (*PUERARIA MONTANA)* INFESTATIONS USING REMOTE SENSING AND
THE NORMALIZED DIFFERENCE VEGETATION INDEX

By

Faye Peters
B.A., University of New Mexico, 2007

A Thesis Approved on

April 18, 2016

By the following Thesis Committee:

_____

Thesis Advisor
Dr. Forrest R. Stevens

_____

Dr. Andrea E. Gaughan

_____

Dr. Margaret M. Carreiro

ACKNOWLEDGEMENTS

Thank you to the University of Louisville Geography and Geosciences Department for providing me the opportunity to be part of a welcoming and helpful academic community that cares about all its students. Thank you to Dr. Forrest Stevens for all his help and patience walking me through the thesis process. I now have a much greater appreciation for the power of computer programming and the opportunities provided by advancements in big data processing. Thank you to my co-advisor Dr. Andrea Gaughan for introducing me to the wild world of remote sensing and her comments that strengthened this thesis. Dr. Margaret Carreiro thank you for being my outside mentor and sharing a passion for invasive species with me. Thank you to Dr. Carol Hanchette and Dr. Keith Mountain for the warm welcomes and guidance throughout my two years in the department. Mr. Robert Forbes, thank you for all your assistance with anything GIS. I would like to send a special thank you to my boyfriend Ben Lacy for his unwavering belief in me and for being a patient critic of all the PowerPoint presentations he got to watch. Thank you to my dad, mom, sister, BB, Bobby and the rest of my family and friends for always supporting me and being there when I need them.

ABSTRACT

ANALYZING THE PHENOLOGIC DYNAMICS OF
KUDZU (*PUERARIA MONTANA)* INFESTATIONS USING REMOTE SENSING AND
THE NORMALIZED DIFFERENCE VEGETATION INDEX

Faye E. Peters

April 18, 2016

Non-native invasive species are one of the major threats to worldwide ecosystems. Kudzu (*Pueraria montana*) is a fast-growing vine native to Asia that has invaded regions in the United States making management of this species an important issue. Estimated normalized difference vegetation index (NDVI) values for the years 2000 to 2015 were calculated using data collected by Landsat and MODIS platforms for three infestation sites in Kentucky. The STARFM image-fusing algorithm was used to combine Landsat- and MODIS-derived NDVI into time series with a 30 m spatial resolution and 16 day temporal resolution. The fused time series was decomposed using the Breaks for Additive Season and Trend (BFAST) algorithm. Results showed that fused NDVI could be estimated for the three sites but could not detect changes over time. Combining this method with field data collection and other types of analyses may be useful for kudzu monitoring and management.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Invasion of non-native plant species has had negative impacts on the environment and cost the United States upwards of $120 billion to eradicate and control (Callen and Miller 2015; Hawthorne et al. 2015). Kudzu (*Pueraria montana)* is a woody, deciduous vine that has become one of the most invasive non-native species in the United States. It was introduced from Asia in 1876 at Philadelphia's Centennial Exposition in the form of seeds and marketed as a way to help with soil erosion in the southeastern region of the United States (Blaustein 2001).

Kudzu exhibits a diverse array of physiological traits that have adapted it to a wide range of climates. None the less, it does prefer certain conditions over others. Its bioclimatic requirements include at least 100 cm of precipitation per year and temperatures between 25°C and 30°C for maximum photosynthesis (Forseth et al. 2004; Lindgren et al. 2013). During peak periods of growth, this species has been known to elongate up to 1 foot per day (Smith 2010). It also has an extensive root system designed for optimal moisture and nutrient extraction. These growth habits often result in kudzu outcompeting other vegetation as it blankets any surface that it encounters. Growth can be stunted when exposed to low temperatures and decreased rainfall (Lindgren et al. 2013).

Many of the 32 states listed as containing kudzu infestations by the Early Detection and Distribution Mapping System (EDDMapS) (EDDMapS 2016) have dedicated

managers monitoring the infestations, and in Kentucky this is overseen by The Kentucky Nature Preserves and Natural Areas (J. Bender, personal communication, 18 September 2015). This organization utilizes EDDMapS to track new and established kudzu infestations. With the help of public citizens, government agencies, and private companies, infestation data is uploaded to this site as point locations along with associated metadata. Because these data are volunteered it may be biased towards accessible and well frequented areas.

The Illinois Department of Natural Resources, uses airborne imagery after the first frost to analyze the extent of infestations throughout the state (J. Shimp, personal communication, 30 September 2015). This is an efficient method because kudzu foliage senesces after the first freeze, thus distinguishing it from other vegetation as seen in Figure 1.

Figure 1. Approximate boundary of a kudzu infestation in Illinois after the first freeze. Courtesy of Jody Shimp IDNR-Division of Natural Heritage

Research Objectives and Hypotheses

The positive correlation of fluctuations in kudzu biomass with the rate of photosynthesis via vegetation indices (Zhitao et al. 2014) builds the foundation of the research question conceived for this thesis project: *"Can the analysis of vegetation indices from remotely sensed data be used to detect the spread and intensity of kudzu infestations?*". The objective and associated hypotheses related to this broad question are as follows.

1. Assess if the density of kudzu's planophile leaf structure inhibits using NDVI for studying its phenology since NDVI has been known to saturate in high LAI areas. *Hypothesis 1: Despite the high leaf area index values of kudzu, NDVI will not saturate at peak productivity, making NDVI a useful index for studying kudzu phenology.*
2. Examine the robustness of fusing Landsat and MODIS imagery to create a time series of NDVI better suited to studying kudzu phenology. *Hypothesis 2: Spatial extents and phenological dynamics of kudzu are better captured by 30 m, Landsat-scale pixels, and higher temporal resolution of MODIS data, making STARFM Landsat-MODIS fused data better than either Landsat or MODIS data on their own for studying kudzu.*
3. Assess the ability of vegetation indices as a reliable method to detect vegetation changes associated with kudzu infestation. *Hypothesis 3: As kudzu infestations expand and intensify measurable vegetation phenology changes. These changes can be detected and measured using BFAST decomposition of NDVI time series.*

Remote Sensing

Remote sensing is the collection of data from a distance and an array of options are available when attempting to map attributes of non-native invasive species (Jensen 2016). Different sensors encompass varying spatial, spectral and temporal resolutions, which require consideration as to which is the most appropriate for mapping non-native invasive species (Cheng, Tom, and Ustin 2007; Hunt, Hamilton, and Everitt n.d.).

Commonly, aerial missions are employed to track infestations because of the availability of high spatial and spectral resolution data collected from airborne platforms. Aircraft are often outfitted with sensors like the hyperspectral Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). It is able to detect spectral signatures in 10nm band increments between 380- 2570nm (Asner et al. 2008; Huang and Asner 2009). Collecting continuous data across the electromagnetic spectrum in hundreds of bands provides the opportunity to assign unique spectral signatures to individual species. When combined with the sensor's moderate to fine resolution, 1 m to 20 m, an accurate representation of vegetation cover might be made. This approach has been used to detect kudzu with an 83.02% accuracy when validated with field data. The authors found that kudzu reflects the highest in the spectrum around 1100nm (Cheng, Tom and Ustin 2007). Drawbacks to using this type of data is that it has a low temporal resolution due to the availability only when a flight path has been tasked and the high cost of operation (Huang and Asner 2009).

Spaceborne multispectral platforms, such as MODIS and Landsat (Buheaosier et al. 2003; Huang & Asner 2009) have been in use for decades monitoring vegetation. Multispectral sensors differ from hyperspectral in that they have considerably fewer

bands across the electromagnetic spectrum over which data is collected. This prevents moderate resolution sensors from spectrally detecting individual plant species due to the spatial and spectral limitations.

The Landsat program began on July 23, 1972 (Jenson 2016) with the launch of Landsat 1 Multispectral Scanner (MSS) to monitor global agricultural practices. Since the maiden launch, sequentially named Landsat satellites have been put into service with the latest being Landsat 8 with the Operational Land Imager (OLI). Landsat 6 was launched in 1993 but did not achieve orbit. Aside from some sensor-to-sensor variation in spectral characteristics, developers have upheld the fundamental goal of the Landsat program which is to retain compatibility across sensors. Changes have included the addition of new bands, such as the coastal/aerosol band to Landsat 8, or band narrowing to assist in spectral differentiation (Jensen 2016). Landsat data was made free and publically available in 2008 through the United States Geological Survey (Wulder et al. 2012).

Landsat has a moderate spatial resolution of 30 m for all bands besides the panchromatic and thermal. The swath width is 185 km which provides ample representation of infestations on the surface (Jensen 2016). The temporal resolution of Landsat data is relatively coarse with a 16 day return time which is often extended due to cloud cover. In this study, climate in eastern Kentucky is prone to rain and snow showers in the spring and winter months (Hill and Mogil 2012) which add to the number of cloudy days. Consequently, a fast growing plant like kudzu would not be sufficiently monitored at a 32-day, or more, temporal resolution making Landsat inferior for detection purposes.

The Moderate Resolution Imaging Spectroradiometer (MODIS), like the Landsat sensors, has been used extensively in the mapping of land surface phenology but is often constrained to the global scale due to the sensor specifications (Muchoney et al. 2000; Zhang et al. 2004; Chuvieco et al. 2013). It has been operational on the NASA based *Terra* and *Aqua* satellites since 2/24/2000. This is a sun-synchronous satellite with a swath width of 2330km and is sinusoidally projected. The fine temporal resolution of one day makes it an ideal candidate for tracking vegetation changes. It is limited by the 250-500 m resolutions available on the bands related to land surface cover. If used on a localized scale there is almost certainly going to be the inclusion of mixed pixels in heterogeneous landscapes.

Huang and Asner (2009) suggest that to successfully use Landsat or MODIS sensors for invasive species detection an infestation site should be expansive and exhibit a phenology different than its surroundings. Privet (*Ligustrum* spp.) and honeysuckle (*Lonicera* spp.) are both non-native shrubs that have been successfully mapped using Landsat and MODIS (Salajanu and Jacobs 2009). These species have the phenological trait of dropping their leaves after most native deciduous trees which aids in distinguishing them from surrounding vegetation. In contrast, when non-native invasive species are obstructed by native vegetation Landsat and MODIS sensors are not as reliable (Huang and Asner 2009). One potential solution to studying localized and/or heterogeneous vegetation cover, which exhibits variation on a spatial scale too fine for MODIS but too rapidly for Landsat detection, is to use both datasets with the use of a data fusion method.

Vegetation Indices

Kudzu is prone to overtake forest canopies, fields and any other surface it comes in contact with, which provides opportunities to be remotely detected. Vegetation indices (VIs) are utilized in the detection process (Blaustein 2001) because they provide a better representation of phenological changes verses basic spectral signatures, i.e. the green spectral signature. Slope based VIs, (Silleos et al. 2006), are commonly used for the detection of phenologic changes to vegetation and originate from the spectral "simple ratio" (SR), which is defined by the inverse relationship between the reflectance (ρ) of the visible red and near infrared (NIR) portions of the electromagnetic spectrum (Birth and McVey 1968; Huete et al. 1997; Turner et al. 1999; Jensen 2016) .

$$SR = \frac{\rho_{RED}}{\rho_{NIR}} \tag{1}$$

Healthier plants exhibit higher photosynthetic rates which results in a higher absorption in the visible red portion of the spectrum and higher reflectance in the NIR portion. This relational difference decreases in unhealthy or senescent plants. VIs are positively correlated with vegetation characteristics such as above ground biomass (Silleos et al. 2006), net primary productivity (NPP) (Rafique et al. 2016) and the level of leaf area index (LAI) (Jin and Eklundh 2014).

Invasive species are most easily detected via remote sensing when they possess a trait that exhibits reflectance properties that are different than their surroundings. Attributes may include earlier green-up time, leaf shape, or flowers (Hunt, Hamilton, and Everitt, n.d.). Kudzu has a noticeably higher LAI value and above ground biomass 10-15 times greater than deciduous forests and other planophile species (Forseth and Innis 2004;

Lindgren et al. 2013). The planophile orientation of kudzu leaves allows it to reflect more in the NIR spectrum and absorb more in the red spectrum when compared to leaves that have an erectophile orientation (Turner et al., 1999). Since VIs are correlated with LAI and above ground biomass they provide a possible means of delineating kudzu from its surroundings.

Examples of VIs are the enhanced vegetation index (EVI) and the normalized difference vegetation index (NDVI). EVI is tailored to mask out atmospheric and soil noise which helps to reduce saturation of vegetation with high LAI values (Huete et al. 1997) . NDVI was first developed by Rouse et al. (1974) and has proven to be a reliable method for extracting phenologic trends across a wide variety of vegetation types. NDVI as a phenology indicator has been used to monitor vineyard growth (Johnson et al. 2003), Mediterranean forest monitoring (Maselli 2004) and the extent of *Lonicera mackii* growth in Cherokee Park, Louisville, KY (Shouse, Liang, and Fei 2013).

Like the SR, NDVI relies on the relative difference in reflectance of the visible red and NIR portions of the spectrum, Equation 2. It differs from the SR in that it normalizes the output value range to -1 to 1, eliminating the effects of having a zero in the denominator.

$$NDVI = \frac{\rho(nir) - \rho(red)}{\rho(nir) + \rho(red} \tag{2}$$

Normalization works by calculating the difference between the visible red and NIR bands which is then divided by the sum of the two values. Reducing the value range decreases the overall effect that the "Soil Line" has on NDVI values as a whole (Silleos et al. 2006). Compiling long term NDVI values into a time series has been found to be a useful

method for detecting abrupt and gradual changes in vegetation over a long time period caused by many types of land cover disturbances.

Google Earth Engine

In 2008, after the USGS release of Landsat imagery, Google launched the cloud-based IDE (Integrated Development Environment) Google Earth Engine (GEE) to run the Earth Engine API (Application Program Interface), also commonly referred to as the Playground. JavaScript is the language of choice within this API. This does require the user have a background in JavaScript as this is not a GUI platform. GEE imagery includes the entire Landsat and MODIS catalogs as well as additional datasets (Padarian, Minasny, and McBratney 2015). All processing and computations are done on the fly (Hansen et al. 2013) which allows the computer to reproject and process data in close to real time. This process automatically does this for any available dataset on the global scale. Geospatial data can be converted to a fusion table, Google's method of geospatial data management, and then loaded into GEE to interact with other vector or raster data. Outputs from GEE analyses are able to be downloaded as georeferenced raster data.

Datasets with highly dimensional spatial and/or temporal resolutions are cumbersome and time consuming for desktop computers to acquire, process and export. GEE's thousands of computers wired in parallel and large collection of data make analyzing big data 40 – 100 times faster than a desktop computer (Padarian, Minasny, and McBratney 2015).

Although the provided data collection is large there are many dataset not available which has been noted as one of the platform's drawbacks. Hyperspectral data is among

the unavailable datasets which limits studies requiring precise spectral classification such as those related to population (Patela et al. 2015), geology (Padarian, Minasny, and McBratney 2015) and forest (Hansen et al. 2013). Personal datasets can be upload to the Playground to offset this limitation but are limited by the 10gb available storage (Padarian, Minasny, and McBratney 2015).

Despite limitations, GEE has been used extensively and proven that a cloud-based remote sensing platform is necessary for research based on highly dimensional datasets. Patel et al. (2015) were able to classify urban extent on the global scale using the normalized difference spectral vector. Maize and soybean crop simulations from the United States were applied to Landsat data acquired via GEE to predict yields (Lobell et al. 2015). A break through global forest map was created by Hansen et al. (2013) that now used by Google to highlight the capabilities of GEE.

Data Fusion Modelling

The spatial and temporal adaptive reflectance fusion model (STARFM), (Gao et al. 2006), generates synthetic imagery by fusing Landsat data with MODIS data. By combining both datasets into one synthetic version so that both high temporal resolution and fine/moderate spatial resolution may be preserved (Gao et al. 2006; Zheng and Moskal 2009). The combination of both resolutions produces an environment that is robust enough to track phenologic productivity changes in vegetation cover while working with a spatial resolution fine enough for smaller, patchy invasive contexts (Gao et al. 2006; Walker et al. 2012; Wang, Gao, and Masek 2014; Schmidt et al. 2015).

Landsat and MODIS data work well for data fusion because they have overlapping spectral bandwidths (Table 1) and similar flyover times which were developed to be comparable (Walker et al. 2012). This method has accurately tracked changes in vegetation dynamics on west Texas ranches (Yang et al. 2015) and after forest/grassland disturbances (Hilker et al. 2009; Schmidt et al. 2015).

TM Bands	TM Bandwidths (nm)	ETM+ Bands	ETM+ Bandwidths (nm)	OLI Bands	OLI Bandwidths (nm)	MODIS Bands	MODIS Bandwidths (nm)
1	450-520	1	450-515	2	450-515	3	459-479
2	520-600	2	525-605	3	525-600	4	545-565
3	630-690	3	630-690	4	630-680	1	620-670
4	760-900	4	750-900	5	548-885	2	841-876
5	1550-1750	5	1550-1750	6	1560-1660	6	1628-1652
7	2080-2350	7	2080-2350	7	2100-2300	7	2105-2155

Table 1. Comparison of Landsat and MODIS spectral bandwidths

This fusion technique works by identifying two base pairs of Landsat and MODIS imagery across a time series that have the same or close to the same acquisition date to create a synthetic image of missing or bad Landsat data within a scene. A significant amount of overlapping, valid data realistically should be shared between image pairs in order to establish a relationship that can be used to generate synthetic imagery (Gao et al. 2006; Wang, Gao, and Masek 2014). Acceptable base pairs are chosen from a sliding window that temporally moves through the times series from iteration to iteration until two sound pairs are chosen. After establishing image pairs, a statistical model is utilized in the prediction of Landsat reflectance or Landsat-scale product derived from MODIS inputs. Predictions produced by the statistical model are used to match unpaired MODIS image dates with Landsat-scale images.

Data introduced into the STARFM algorithm must be corrected and assume some basic principles (Gao et al. 2006). Atmospheric correction and resampling are required before input into the algorithm. First, surface reflectance is estimated prior to fusion processing requiring all MODIS and Landsat imagery to be atmospherically corrected using the same basic principles. Most commonly Landsat Ecosystem Disturbance Adaptive Surface Reflectance (LEDAPS) (Maiersperger et al. 2013; USGS 2016) is used for Landsat images because it is very similar to the correction algorithm applied in the Collection 6s approach for MODIS data (Gao et al. 2006; Maiersperger et al. 2013). Second, both datasets must have the same spatial resolution which requires all MODIS imagery to be resampled to 30 m to match that of Landsat.

An ideal study area to run through STARFM would provide the user with pure pixels, unchanging vegetation extent and uniform spectral reflectance (Gao et al. 2006). In the real world this does not exist and is accounted for in the STARFM algorithm by weighting pixels that neighbor a central pixel. Three factors are considered: spectral difference, temporal difference and location difference. The product of these weights make up the combined weight assigned to a pixel (Equation 4).

$$C_{ijk} = S_{ijk} * T_{ijk} * D_{ijk} \quad (4)$$

Where C is the combined weight, S is the spectral difference, T is the temporal difference and D is the location difference (Gao et al. 2006). i, j, k refers to the x/y coordinates and acquisition date of a pixel, respectively. All three exhibit an inverse relationship between what is measured and how heavily the pixel is weighted. The lower

the measurement of the three variables, the higher the weight assigned to a pixel (Gao et al. 2006).

To increase the efficiency of this product, additional weights can be assigned to pixels based on the level of homogeneity in the study area. Spectral classification of pixels must be examined to ensure that correct weighting is assigned. Classification options include an unsupervised method or utilizing a threshold indicator built into the STARFM algorithm. The difference between the two is that the unsupervised method applies the rules over the entire study area while the second method only applies to the pixels that are of concern. Beyond this step bad pixels can be masked as they can create bias within the statistical predictions used to form the synthetic image.

Temporal Decomposition

NDVI estimates can be combined at the pixel level into time series which can then be decomposed in order to analyze underlying trends associated with phenologic change. Such are required to be relatively long periods, which in remote sensing terms translates to the numbers of images acquired for a distinct subset of land. Time series are classified as having high or low dimensionality depending on the amount of observations. Dimensionality refers to the length and number of temporal observations within a dataset. For example, a dataset consisting of all available Landsat TM images would have high dimensionality versus a low dimensionality dataset consisting of only two Landsat TM images.

Low dimensional time series often use methods such as Principal Component Analysis or Fourier analysis (Verbesselt et al. 2010b). Both are transformation methods

that aim to reduce the dimensionality of a dataset down to the components with the highest variance (Lhermitte et al. 2011). Most often such methods are used to assess change by partitioning variation into various subcomponents including large shifts. Transformation methods reduce underlying trends but typically mask them because they do not contribute a high amount to the total variance.

Datasets with high dimensionality benefit when seasonal and other trends are retained as the data is reduced. Breaks for Additive Seasonal and Trend (BFAST) (Verbesselt et al. 2010a; Forkel et al. 2013) does this by decomposing a time series into seasonal, linear and error components as illustrated in Figure 2. This means that all the decomposed parts when summed reconstruct the observations through time (Equation 3).

$$Yt = St + Tt + et \tag{3}$$

Where *Yt* is the observed trend, *St* is the seasonal trend component, *Tt* is the linear trend component and *et* is the error component.

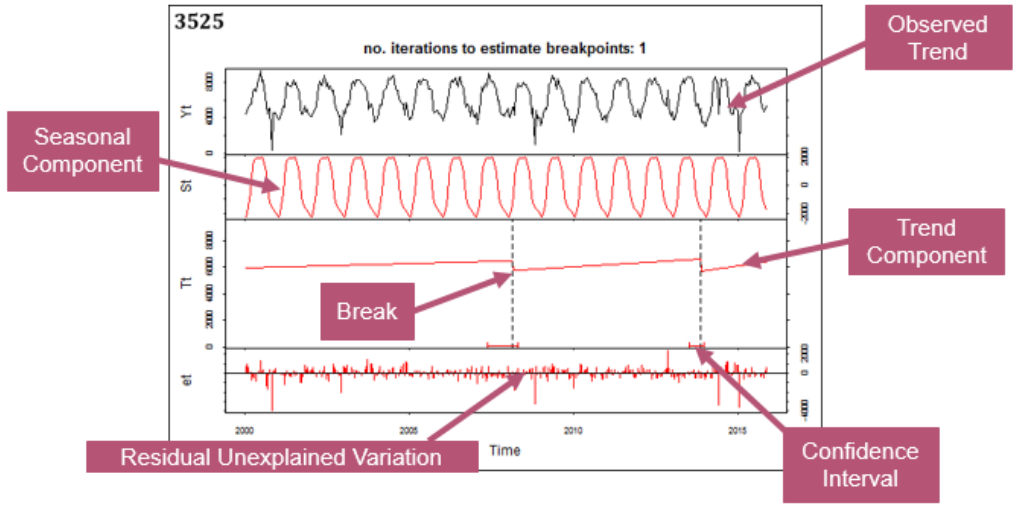

Figure 2. Example BFAST Plot

A harmonic analysis is applied to decompose the observed seasonal trend from other more predictable parts of the series. This type of analysis is more robust than the "dummy" model method because it uses a continuous baseline for decomposition rather than discrete points (Verbesselt et al. 2010b; Hutchinson et al. 2015). Discrete points chosen for the dummy model express phenologically important dates such as green up or leaf drop thus obscuring what occurs between these chosen points. The combination of multiple sinusoidal waves comprise a harmonic analysis which varies based on changes to the amplitude and phase of the wave (Jakubauskas, Legates, and Kastens 2001).

Abrupt or gradual changes in vegetation can be detected after decomposition by applying breaks to the linear trend to better fit detected errors. This is accomplished by analyzing any remaining trends leftover from the initial decomposition process (Hutchinson et al. 2015; Verbesselt et al. 2010a) by specifying the acceptable number of breaks that can be added to the linear trend. A set number of iterations is chosen for the BFAST algorithm to complete before selecting the best fit for breaks along the linear trend. The linear trends connecting breaks are useful in characterizing duration of disturbances, as well as the nature of vegetation dynamics pre- and post- break. A level of uncertainty bounds the breaks meaning that the true date of disturbance may lie before or after the break date. These break dates correspond to remote sensing acquisition dates which will vary based on the temporal resolution of the sensor being used.

CHAPTER 2: SIGNIFICANCE

The objective of this project is to produce a method incorporating satellite imagery that management officials can use to assess kudzu (*Pueraria montana*) infestations. The study sites were chosen in Kentucky because of the location along the northern edge of the North American kudzu extent and the presence of smaller, localized infestations. Monitoring and management methods should display that that they are effective at assessing the kudzu phenologic dynamics in a variety of infestation site sizes and be economical to implement.

The Kentucky Transportation Cabinet (KYTC) District 10 is currently working with the University of Kentucky to test chemical and mechanical methods to control kudzu. The current method being used to monitor the effectiveness of these applications is ground-based field collection. This method is labor intensive and may not provide the coverage and scale needed to accurately monitor the efficacy of eradication methods. A method like the one proposed in this study would reduce the labor time and cost as well as compliment the ongoing field-based monitoring system that is currently in place.

The proposed method could also be applied to a broader spectrum of kudzu infestations outside of Kentucky as well as other non-native vine species inside and outside of the state. Vines such as porcelain berry, *Ampelopsis glandulosa*, exhibit characteristic similar to that of kudzu that both threaten ecosystems and are difficult to eradicate and or control.

CHAPTER 3: METHODS AND DATA

Landsat and MODIS imagery was incorporated into the STARFM data fusion algorithm to produce the most accurate series of images to apply the BFAST temporal decomposition model to. This process aimed to produce a dataset with sufficient temporal and spatial resolution to assess productivity change using decomposed NDVI time series of kudzu infestations in three study sites. Similarities between both datasets and their accessibility through open source data archives made Landsat and MODIS an ideal choice for this project. Google Earth Engine (GEE) was employed to acquire, process and export both sets of imagery because of its immense on the fly processing power of individual Landsat and MODIS scenes. The RStudio interface provided a platform to run both the STARFM and BFAST algorithms which fused and temporally decomposed the imagery, respectively.

Study Area

Infestation sites were chosen based on a set list of criteria. According to the United States Department of Agriculture (USDA) the smallest infestation site should be least 4 times as large as the image pixel (Hunt, Hamilton, and Everitt n.d.). The 30 m Landsat resolution determines that the smallest infestation should be larger than 3,600 m^2.

Study areas were supplied by the KYTC District 10 (D. R. Gumm, personal communication, 29 October 2015) and are clustered in the eastern portion of Kentucky in Breathitt and Perry counties (Table 2 and Figure 3). All sites were delineated using Google EarthPro™ on 6/13/2014 and saved as a .kml file which was later converted to a .shp file using ArcMap 10.3.1.

Site	Latitude	Longitude	Area (m^2)	Elevation (m)	Site Location
#1	37°32'8.26"N	83°32'36.25"W	52,936 (13 acres)	281.94	KY 2469 Athol, KY
#2	37°13'13.83"N	83°10'37.72"W	34,459 (8.51acres)	369.72	KY1096 Hazard, KY
#3	37°12'11.84"N	83°11'33.10"W	28,777 (7.11acres)	541.02	KY1096 Hazard, KY

Table 2. Specific Location of the three study sites.

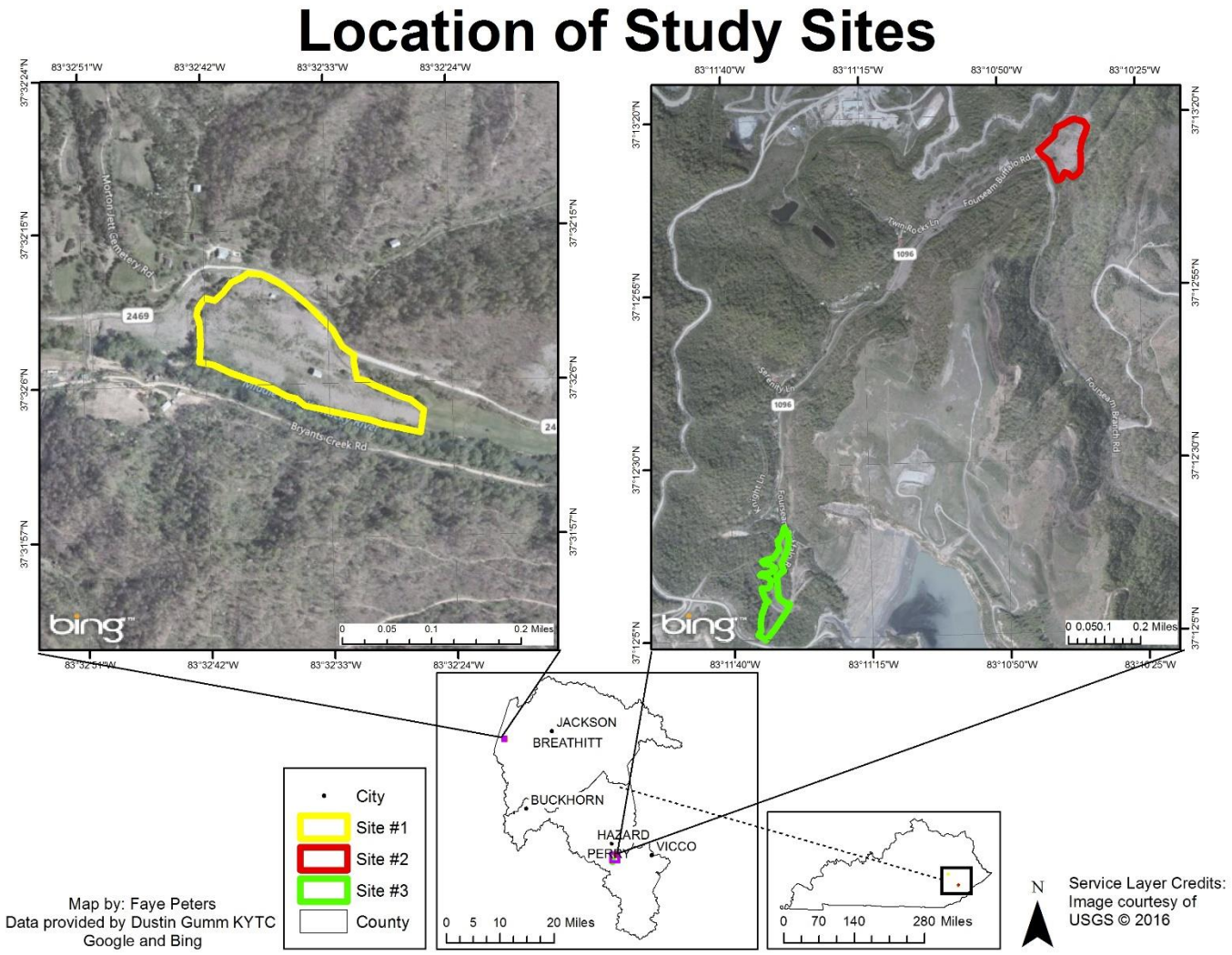

Figure 3. Location of kudzu infestation study sites in eastern Kentucky.

The study areas are located in the Cumberland Plateau region of eastern Kentucky. Forested, rolling hills intertwined with rivers and creeks cover this landscape. The actual sites sit at lower elevations (Figure 4) than the surrounding landscape and do not exhibit one slope aspect over another.

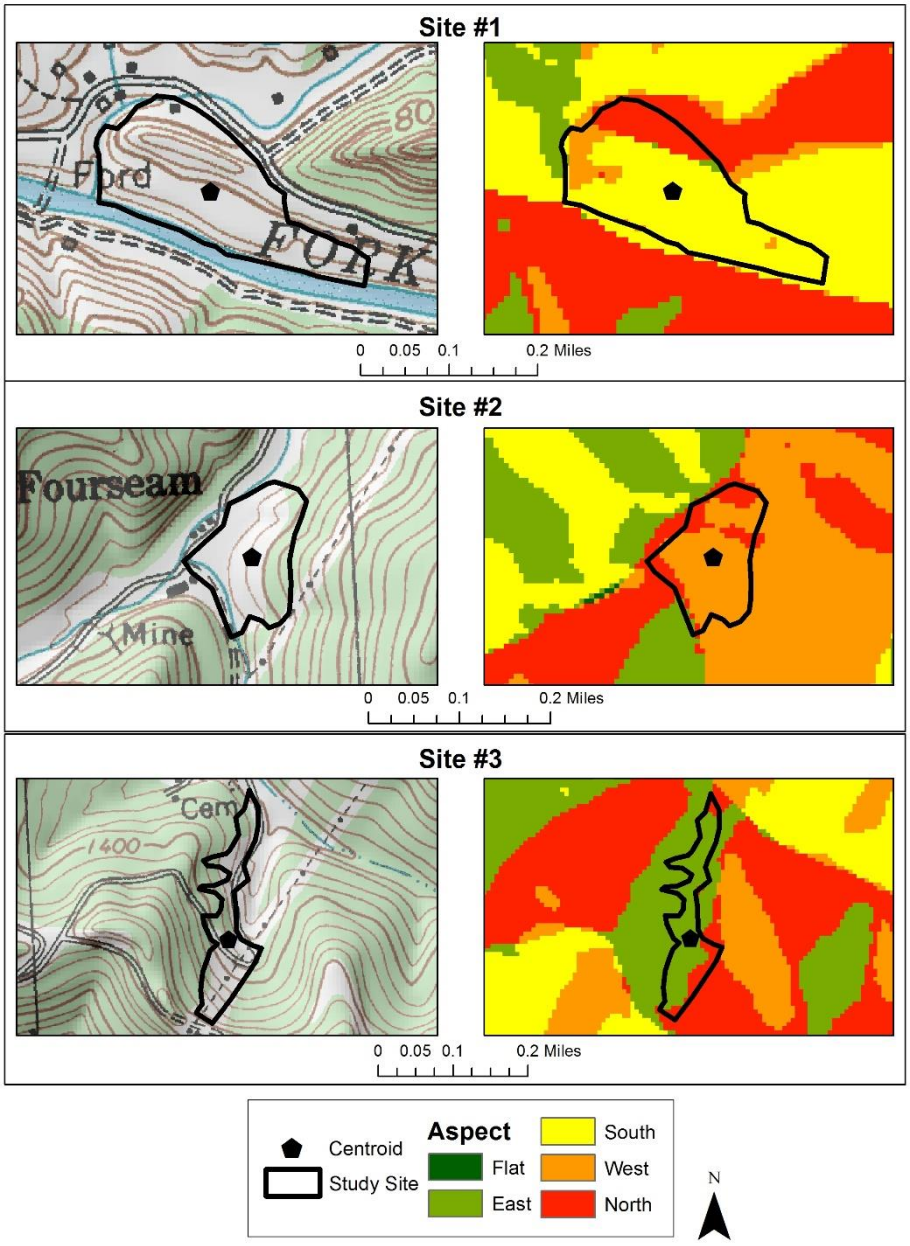

Figure 4. Slope and aspect of the three study areas.

Disturbed edges are prime environments for kudzu to thrive due to greater sunlight for photosynthesis and other factors (Blaustein 2001; Smith 2010; Lindgren et al. 2013). Disturbance sources include, the eastern Kentucky coal fields, which can be clearly seen in images from Perry County. Images acquired from Google Street View display the robust and aggressive nature of this vine (Figure 5 and Figure 6).

Figure 5. Site #2 kudzu infestation KY 1096 looking southeast.
Image: Google EarthPro™ (4/10/2016)

Figure 6. Site #3 kudzu infestation KY 1096 looking north.
Image: Google EarthPro™ (4/10/2016)

Centroid pixels were used to compare productivity trends between the three sites to determine whether a NDVI time series "signature" exists for kudzu infestations like those commonly encountered in Kentucky. Centroids were calculated in ArcMap 10.3.1 using the Feature to Point tool in the Data Management Toolbox. The centroid for Site #3 was shifted south to ensure that the pixel was completely contained within the site. NDVI values from the pixel containing the centroid from the MODIS and synthetic data were used to generate the time series and subsequent BFAST temporal decompositions.

A series of random points were created in each study area to determine if the trends detected near the centroids were representative of the entire area or if it was an isolated occurrence. These extra points were generated randomly using the Create Random Points tool within the ArcMap Data Management toolbox as illustrated in Figure 7. Five points were selected with a minimum allowable distance of 30 m. Before running the BFAST on any of these points it was made sure that none were located within the same 30 m pixel or in a pixel not completely contained within the study area.

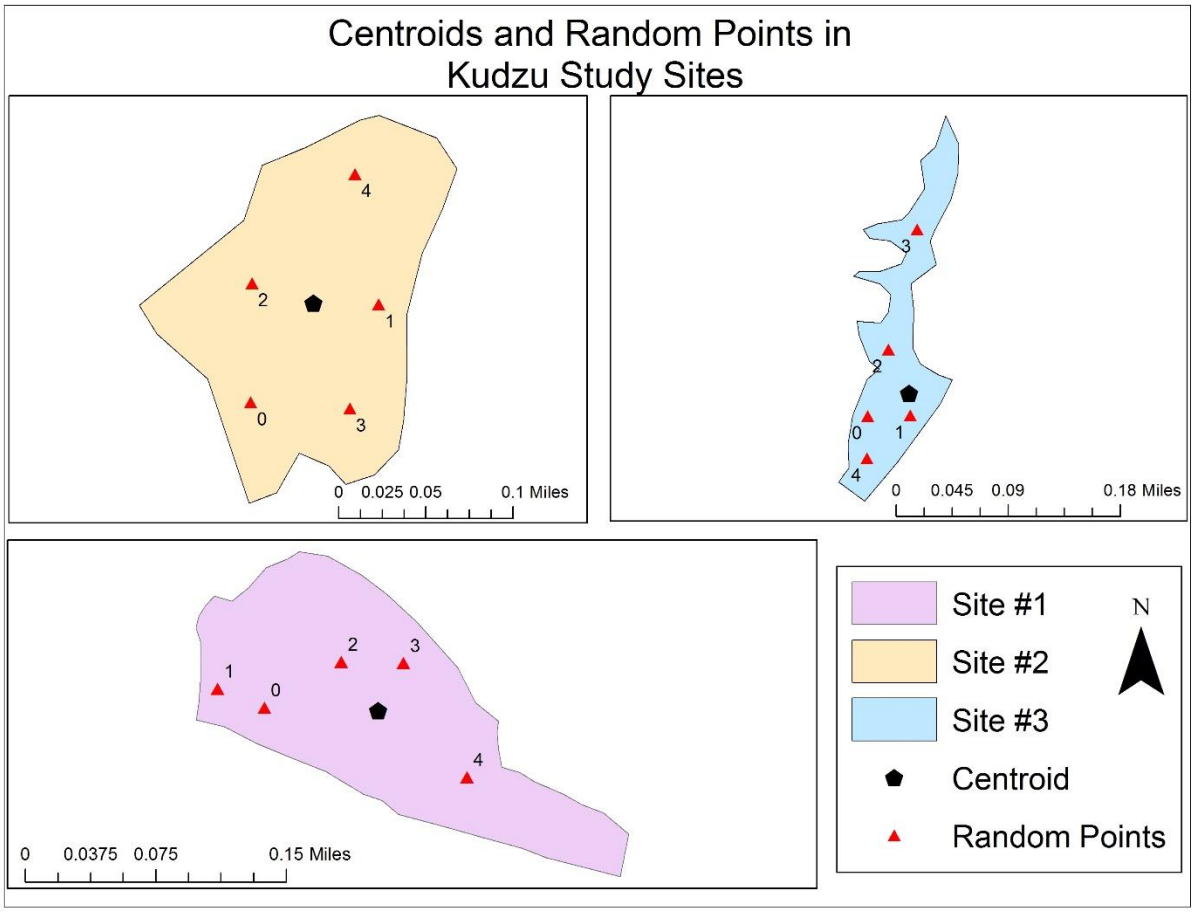

Figure 7. Placements of centroids and random points in each of the study sites.

Datasets

Both Landsat and MODIS datasets were acquired and processed using the GEE, cloud-based API. The JavaScript-based GEE interface script that subset, resampled and exported all MODIS and Landsat data used in the study is included in Appendix A. All Landsat images were acquired from the GEE archive where surface reflectance estimates were calculated on the fly via the LEDAPS algorithm for dates ranging from 2/18/2000 to 12/31/2015 along Path 19 Row 34 (Table 3). This time frame was chosen to coincide with all possible MODIS imagery through the end of 2015.

SENSOR	ACQUISITION DATES
Landsat 5-TM	3/12/2000 to 11/6/2011
Landsat 7-ETM+	10/1/2007 to 12/31/2007 4/1/2011 to 6/30/2013
Landsat 8-OLI	4/11/2013 to 12/31/2015

Table 3. Landsat Image Acquisition Dates

Landsat 7 ETM+ experienced a scan line corrector (SLC) failure on May 31st, 2003 (Gu and Wylie 2015) resulting in images with strips of data missing due the sensor not being able to compensate for movement during data acquisition. Considerations were made to remove these data but would have resulted in a time gap measuring a few months between the SLC off date and the launch of Landsat 8 (Goward et al. 2006) as well as a period in 2007 when TM went offline. Due to the rapid phenological changes of kudzu and the need for continuous data for BFAST to work most efficiently, it was decided to include all ETM+ images for possible matches with MODIS/Landsat image pairs.

The spectral resolutions between Landsat sensors varies slightly, so all Landsat products were resampled to the standard 30 m during the GEE export process (Appendix A) (Irons, Dwyer, and Barsi 2012). The visible red and NIR bands were used to calculate the NDVI for the study areas (Table 4)) at each image date which were then processed into a single multiband image (Appendix A). Cubic convolution resampling was used during image resampling and data was projected to UTM Zone 17N (WGS84) in GEE (Appendix A).

	Red Band	NIR Band	Spatial Resolution
Landsat 5 TM	630 – 690nm	760 – 900nm	30m
Landsat 7 ETM+	630 – 690nm	780 – 900nm	30m
Landsat 8-OLI	640 – 670nm	850 – 880nm	30m

Table 4. Landsat Red and NIR Band Resolutions

The MODIS MOD13Q1 Vegetation Indices 16-Day Global 250 m product was used to construct an NDVI time series. This dataset was collected from the MODIS sensor aboard the *Terra* platform. These data were acquired and processed via GEE for the time period of 2/24/2000 to 12/31/2015 for all three study sites using tile h11v05. These data provide EVI, NDVI and surface reflectance from which NDVI was estimated. Quality assurance mask bands were used for the exclusion of bad pixels below the lowest level of decreasing quality as specified in the MOD13Q1 data description (Appendix A) (LP DAAC 2014). Dates assigned to this image collection correspond to the first day of each 16 day period.

Beginning in 2015 the 6th version of the processing algorithm for the MOD13Q1 dataset was released, which needs to be used for all analyzed data (Gao et al. 2006). This algorithm estimates surface reflectance via atmospheric correction using bi-directional reflectance, and is similar to the Landsat LEDAPS method of atmospheric correction which adds to the comparability between Landsat and MODIS datasets (Walker et al. 2012; Wang, Gao, and Masek 2014)

Google Earth Engine Data Processing

TM, ETM+ and OLI image collections were filtered for only those available within the date range of MODIS images. To reduce the file size of final time series, data from each of the three study areas was acquired and assembled into separate multiband image stacks. A 5000 m buffer was included so that all pixels overlaying the study sites were extracted for both MODIS and Landsat datasets and to provide potential areas to be used in a land cover comparison.

The STARFM algorithm is most efficient when then percentage of bad pixels is kept to a minimum (Gao et al. 2006). For this reason everything besides clear and water pixels were masked from each Landsat image and if that accounted for more than 5% of the total number of pixels then the image was not included in the fusion process. From this point forward the term "bad" in reference to the quality of images used in this study will be defined as those with >5% bad pixels.

As explained in the methods and GEE code (Appendix A), NDVI was calculated for each image collection. Separate lines of code were constructed for the TM/ETM+ and OLI image collections as the band numbering is different between these sensors. After the filtering process was complete, all Landsat images were merged into one Landsat multiband image. Each band contained either good and/or masked pixels.

The merged Landsat collection was matched against the merged MODIS imagery by image date. MODIS NDVI values are stored in GEE as integers which does not correspond with Landsat's floating point estimates of NDVI. Thus, Landsat NDVI values were multiplied by 10,000 to convert them to integer values. At which point, two complete and comparable multiband images were saved for all three study sites for every

MODIS 16-day composite date between February 2000 and December 2015. In addition, summary tables recording the composite dates and percentage of bad pixels for each site were exported and saved.

STARFM Data Fusion

The STARFM data fusion algorithm was used for this study to synthesize the high temporal resolution of MODIS 16-day composites with the higher spatial resolution of Landsat data (30m) (Gao et al. 2006). The STARFM v.1.2.1 algorithm was run from command line tools downloaded from the USDA Agricultural Research Service website (United States Department of Agriculture 2016) but executed using the R statistical environment (R Core Team 2016). The series of STARFM commands used in the study is documented and provided in Appendix B.

Before the implementation of data fusion, a threshold of 5% bad pixels was set to define which MODIS and Landsat pairs would be used in the process. Any pixel within the buffered area or the border surrounding it containing clouds, snow or shadow were masked and considered "bad." Any masked pixel within the 5000 m buffered area was counted and the percentage of the total area these pixel comprised was calculated. If the number of "bad" pixels inside the border region exceeded 5% of the total the image was defined as "bad" and it was not used to create "good" MODIS/Landsat pairs. The total number of images used in the fusion process are shown in (Table 5

Table **5**) and graphically in (Figure 8).

Landsat (TM, ETM+, & OLI) Image Contribution				
Site	Good Images (≤5% Bad Pixels)	Bad Images (>5% Bad Pixels)	Missing Images (All Bad Pixels)	Total Images
#1	76	162	127	365
#2	52	75	238	365
#3	55	79	231	365

Table 5. Breakdown of Landsat Imagery used in the STARFM fusion process.

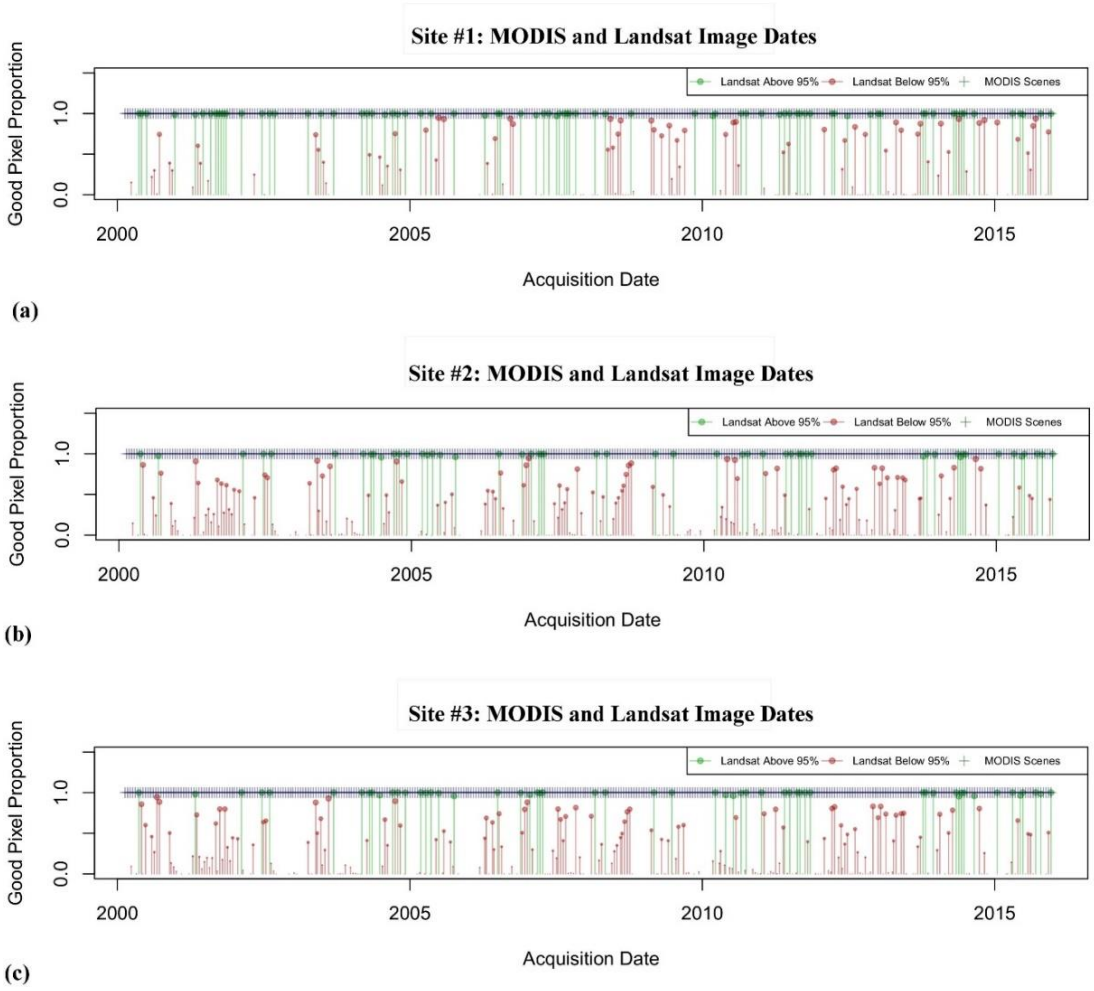

Figure 8. Graphical representation of Landsat images above and below the 5% bad pixel threshold. No percentage correlates to no image. MODIS images along the 1.0 lines representing all good imagery from that collection. Site #1 (a), Site #2 (b), Site #3 (c)

For each MODIS 16-day image date STARFM used a moving window to locate the first Landsat/MODIS image pair containing at least 95% good pixels directly before and after it. The purpose of these image pairs was to simulate a Landsat spatial scale estimate of NDVI at the MODIS image date between the identified paired images. The outcome of this process was a synthetic time series of Landsat-scale data for each MODIS image date. Upon completion of the fusion process, any masked Landsat pixel was replaced with the Landsat-scale synthetic data produced at the corresponding MODIS image date. This method retained the most original Landsat data by compiling a Landsat-scale image stack comprised of original, filled and completely synthetic estimated NDVI images of which was used in the BFAST analysis. This process is featured in Figure9.

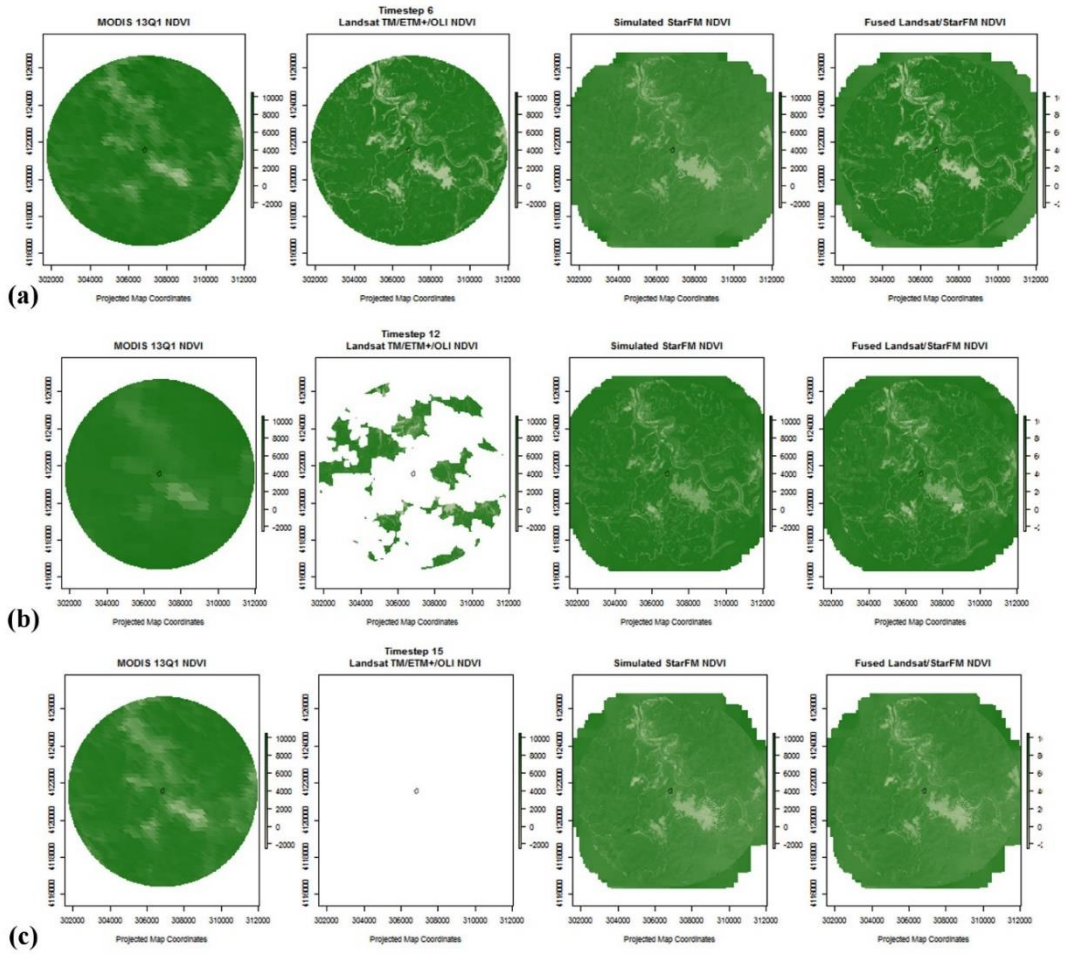

Figure 9. STARFM image comparison from Site #1. From left to right: MODIS, Landsat, Synthetic, Filled. Image sequence (a) time step six which contains Landsat image with no missing data, (b) date twelve contains masked pixels that are filled with synthetic data, (c) date fifteen, which contains no valid Landsat pixels and which is replaced with a complete, synthetic or simulated set of pixels images.

BFAST Temporal Decomposition

Following the fusion process, all non-missing Landsat data was filled in with synthetic, STARFM data to produce a complete time series at a 30 m spatial resolution. The BFAST package for R (Verbesselt et al. 2010a; Verbesselt et al. 2010b) was used to process and decompose the time series for each infestation site, Appendix C. A total of

six pixels were evaluated by compiling their NDVI values into a time series. These included the pixel corresponding to the centroid and five additional pixels for each site. The resulting series of decompositions shows phenological variance and change measured by NDVI (Appendix E).

CHAPTER 4: RESULTS

The fused Landsat time series assembled from Landsat-derived and STARFM synthetic data proved to be a better scale to evaluate the productivity of kudzu in all three study sites compared with one composed strictly of MODIS data. All final Landsat-scale images were combined into an estimated NDVI time series. It was important to derive a complete time series for the time range of this study to avoid any temporal gaps but consequently cloud, shadow, snow, SLC-off errors and other contaminates were included. Figure 0 - Figure 2 highlight the range of gaps in the Landsat data, red line, and how those gaps were replaced with filled and synthetic imagery.

Evaluation of the differences between MODIS and Landsat-scale pixel resolution was made by comparing the results of BFAST plots from MODIS only data and fused time series. From all kudzu study area centroids and points, time series were extracted and decomposed with the BFAST algorithm (Appendices D and E).

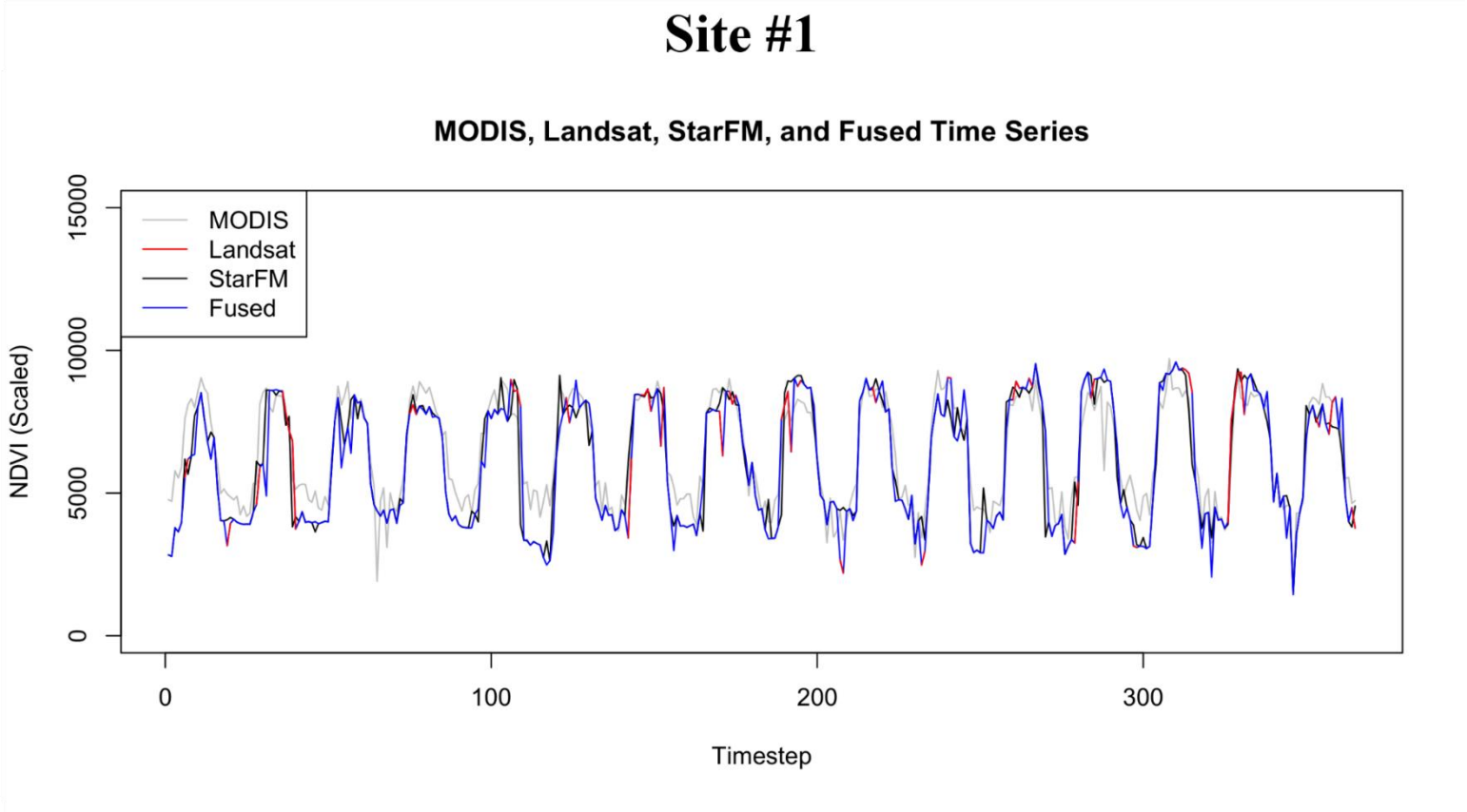

Figure 10. Site #1 times series for MODIS, Landsat, STARFM, and Fused datasets. The red line represent available good Landsat data and the prevalence of gaps in that image collection.

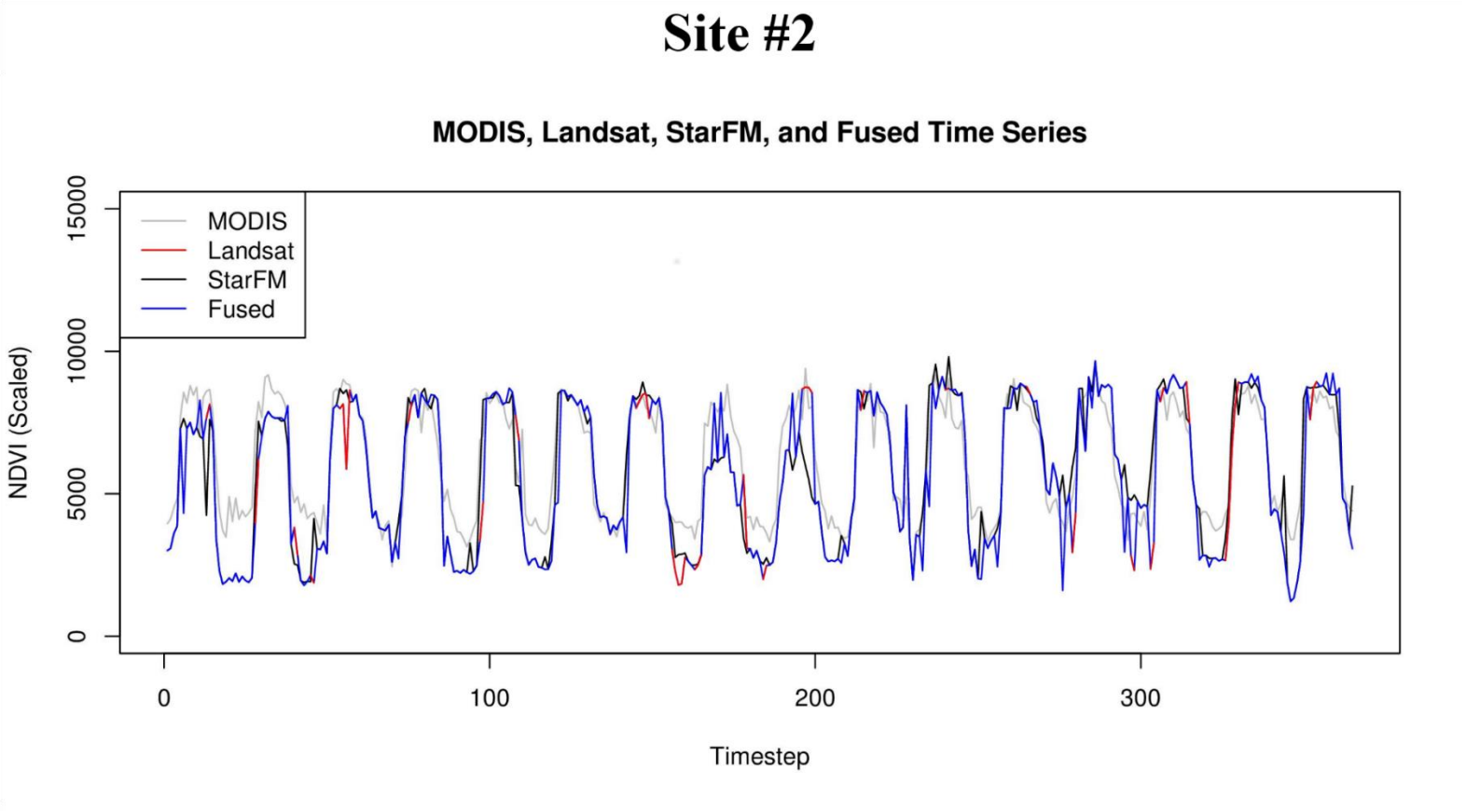

Figure 11. Site #2 times series for MODIS, Landsat, STARFM, and Fused datasets. The red line represents available good Landsat data and the prevalence of gaps in that image collection.

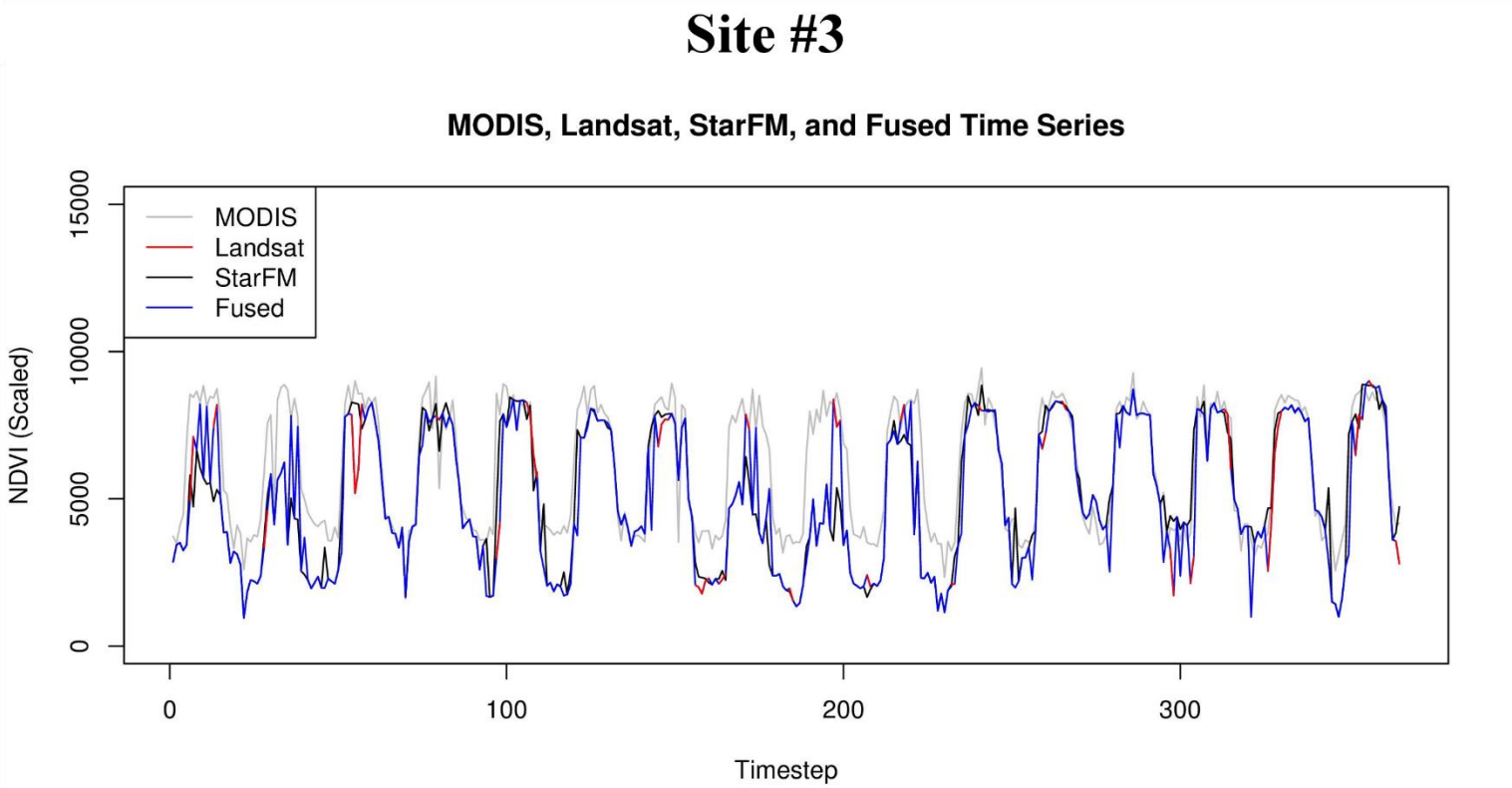

Figure 12. Site #3 times series for MODIS, Landsat, STARFM, and Fused datasets. The red line represent available good Landsat data and the prevalence of gaps in that image collection.

Entropy values for both the yearly and seasonal trends expose the complexity of seasonal trends in the data. The lower the entropy value the more order present in the trend line. Comparisons of BFAST results across sites show minimal difference between locations and sensors. These differences are listed in Table 6.

	Yearly Entropy			Seasonal Entropy		
	Site #1	Site #2	Site #3	Site #1	Site #2	Site #3
(a)MODIS						
Centroid	0.4521	0.3346	0.3638	0.2290	0.2554	0.2511
(b) Landsat						
Centroid	0.4042	0.4040	0.4714	0.2451	0.2412	0.2392
0	0.4662	0.3996	0.4140	0.2474	0.2419	0.2434
1	0.4179	0.4147	0.4440	0.2394	0.2424	0.2430
2	0.3595	0.3787	0.4111	0.2438	0.2422	0.2380
3	0.3880	0.4200	0.4153	0.2512	0.2385	0.2446
4	0.5430	0.4454	0.3795	0.2384	0.2429	0.2464
Average	0.4298	0.4104	0.4226	0.2442	0.2415	0.2424

Table 6. Yearly and seasonal trend line entropy values for (a) MODIS and (b) Landsat-scale, Fused Data time series.

The fused data did exhibit breaks in the seasonal trend (St) at Site #1 which was not observed in the coarser MODIS data. This was observed at time step 97 which correlates with the date 4/22/2004. Unlike the other two sites where the estimated seasonal trend was consistent across time, and similar between MODIS and fused imagery, the identification of this difference could be a signal of disturbance or shift in established vegetation at Site #1. Comparisons between the similar seasonal trends (St) are illustrated in BFAST analyses for each site provided in Figure 13 - 15.

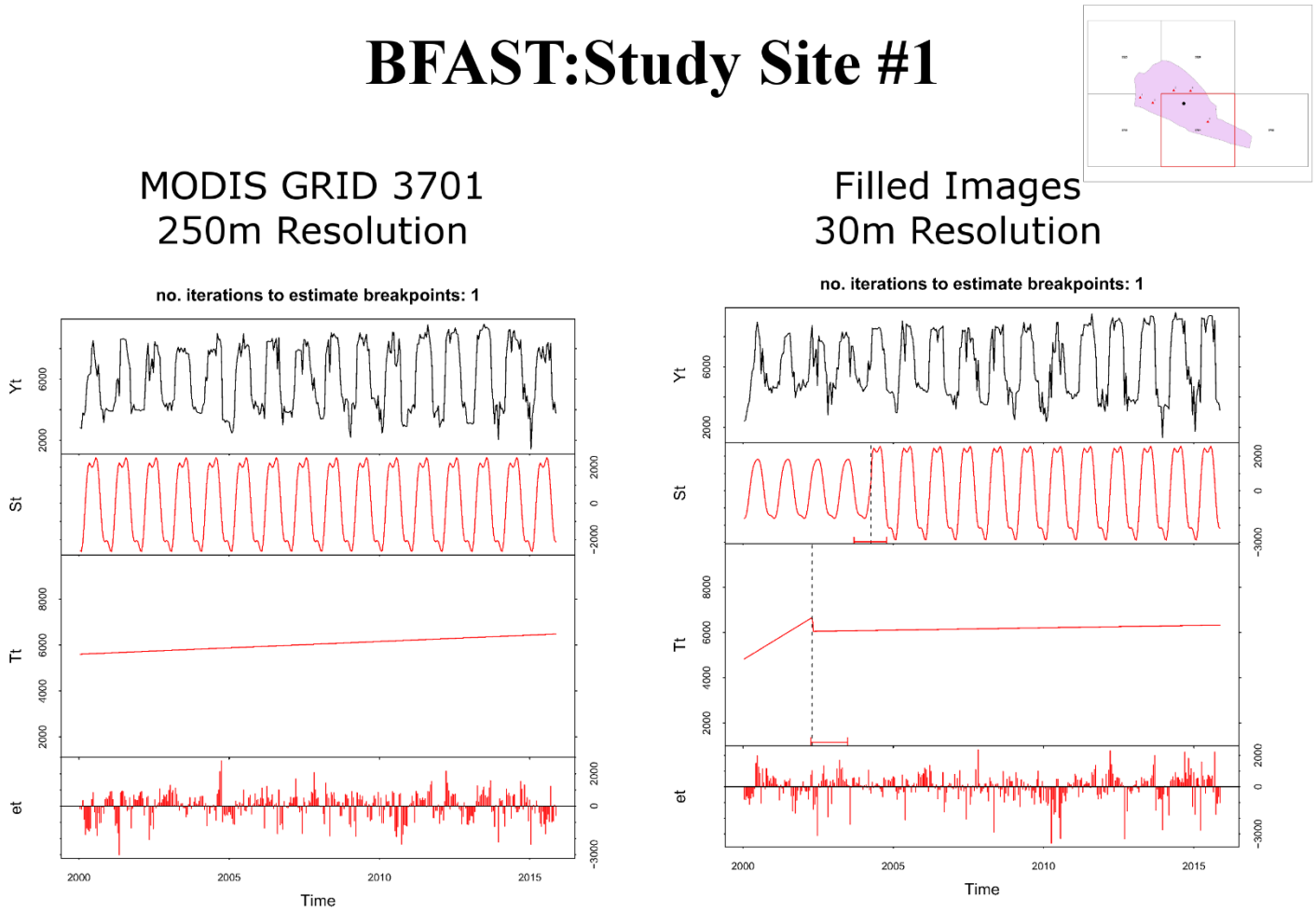

Figure 13. Comparison of the BFAST plots for the MODIS (left) and filled image collection (right) pixels that contain the centroid of Site #1.

BFAST:Study Site #2

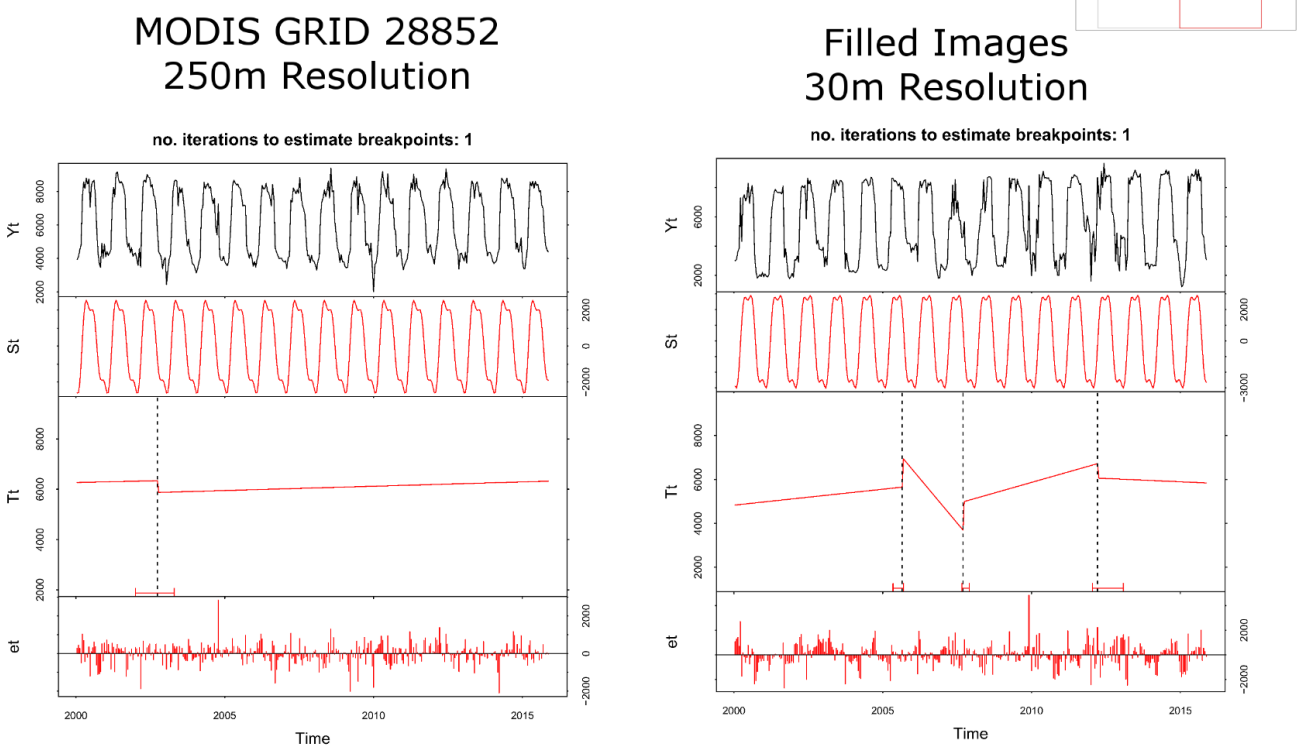

Figure 14. Comparison of the BFAST plots for the MODIS (left) and filled image collection (right) pixels that contain the centroid of Site #2

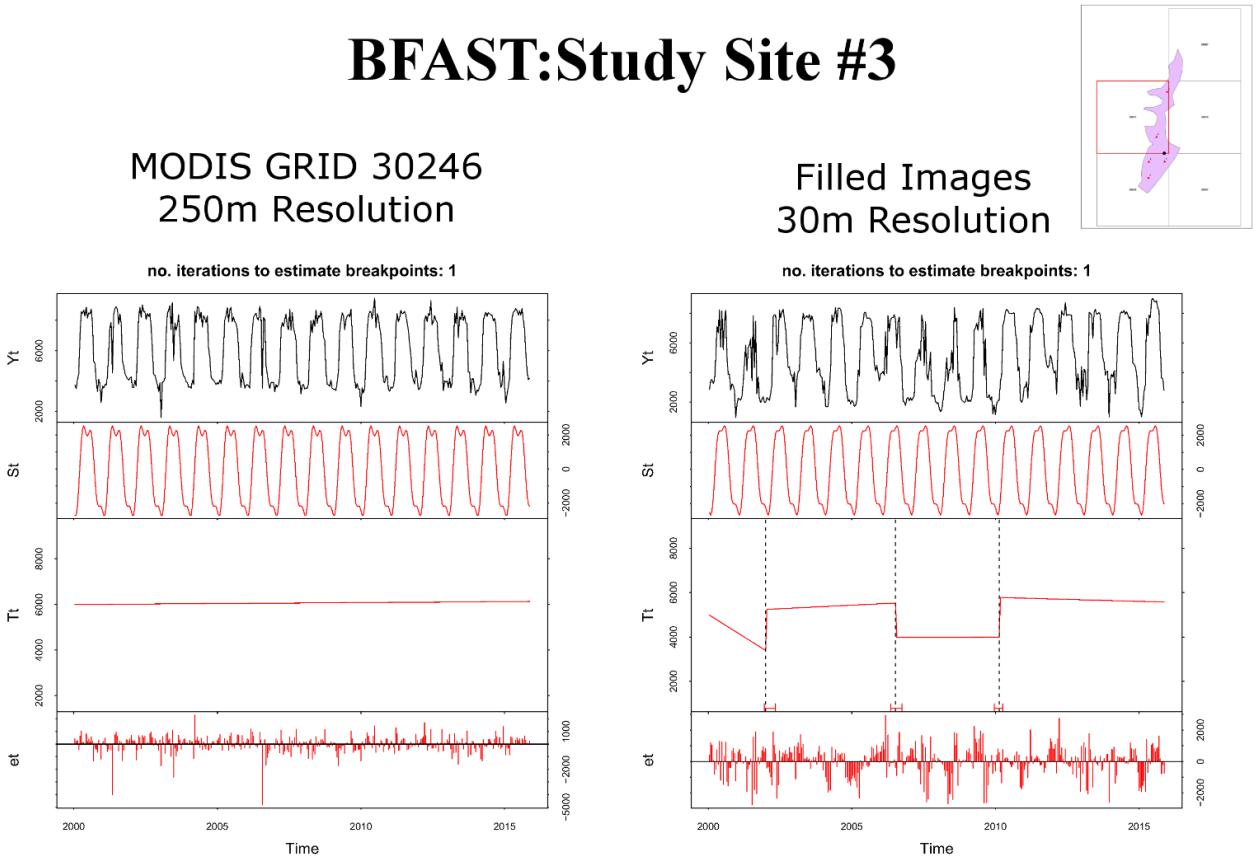

Figure 15. Comparison of the BFAST plots for the MODIS (left) and filled image collection (right) pixels

From the BFAST plots generated at each of the 6 points there are trends within each study area, Appendix E. With the exception of one point, Site #1 exhibited no breaks and a positive linear trend over time. Sites #2 and #3 produced BFAST plots similar to the one produced for each centroid. Three breaks in the long term, linear trend (Tt) are estimated at Sites #2 and #3 with the BFAST algorithm. These sites are located close to one another along Kentucky State Road 1096. Site #2, point 1 exhibited two breaks compared to three for the rest of the points, Table 7. The date of the second break better coincides with the timing of the third break. For this reason this break was moved to the Break 3 column.

With a slight margin of error at the beginning and ending date of each break can be estimated by using the image band located at the break as a reference point. It is approximated because the date that it is referring to is also the date for which each MODIS MOD13Q1 16-day composite is labelled. The break is also approximate with regards to the disturbance or change in phenology that generates it since the effect measured in vegetation index may lag the cause of that change. The first break shows the most consistency between the two sites and was detected at 10/16/2005 as the approximate date highest frequency of same date breaks in NDVI (Table 7). The second and third breaks show a lower amount of overlap that also falls in the later months of the year.

Site #2						
Point	Break 1	MODIS Date	Break 2	MODIS Date	Break 3	MODIS Date
Centroid	131	10/16/2005	178	11/1/2007	282	5/8/2012
0	131	10/16/2005	178	11/1/2007	283	5/24/2012
1	154	10/16/2006	N/A	N/A	318	12/3/2013
2	47	2/18/2002	154	10/16/2006	283	5/24/2012
3	131	10/16/2005	178	11/1/2007	316	10/16/2013
4	154	10/16/2006	237	5/25/2010	283	11/1/2013

(a)

Site #3						
Point	Break 1	MODIS Date	Break 2	MODIS Date	Break 3	MODIS Date
Centroid	47	2/18/2002	151	8/29/2006	234	4/7/2010
0	131	10/16/2005	178	11/1/2007	270	11/1/2011
1	47	2/18/2002	131	10/16/2005	197	8/28/2008
2	131	10/16/2005	197	8/28/2008	287	7/11/2012
3	131	10/16/2005	197	8/28/2008	258	4/23/2011
4	57	7/28/2002	151	8/29/2006	318	12/3/2013

(b)

Table 7. Break location by band and the corresponding break start date based on the MODIS MOD13Q1 16-day composite dates for Sites #2 (a) and #3 (b)

Comparison with Vegetation Outside the Study Area

Beyond the study area are forest stands and fields. To compare any similarities and contrasts that may exist between kudzu and another vegetation cover a simple spectral profile was created using NDVI values for three types of vegetation: kudzu, forest and grass. Site #2 was used for this portion of analysis as results are assumed to be similar for the same types of vegetation near the other two infestation sites. The spectral profile reveals that overall grass shows lower and smaller ranges in NDVI values compared to the other vegetation covers most likely caused by its erectophile leaf structure (Turner et al. 1999). Forest and kudzu have similar entropy patterns with forest cover peaking with slightly higher NDVI values at both peaks and troughs. These variations are illustrated in Figure 16.

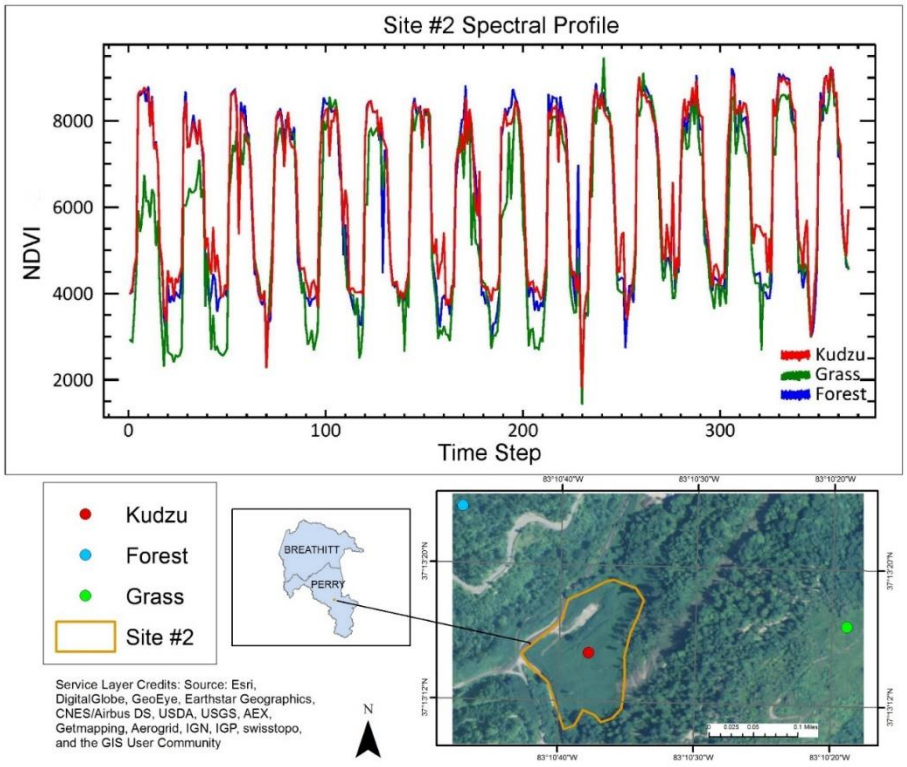

Figure 16. Spectral profile representing kudzu, grass and forest cover in and around Site #2.

CHAPTER 5: DISCUSSION

Results from this study show that original and decomposed time series indicate that estimated NDVI values for kudzu do not oversaturate and can be used as a resource to analyze the phenology dynamics of this plant. Analysis of the results also show that combining imagery into a fused image collection, filling temporal gaps in Landsat-scale NDVI with MODIS-derived, and synthetic data increase the applicability of this type of data to detect long-term trends and changes. Being a fast-growing and high biomass species (Forseth and Innis 2004; Lindgren et al. 2013) we assumed that this method would have successfully detected expansion and intensity of kudzu infestations from the surrounding vegetation. However, the scales used in this study were not fine enough to positively distinguish growth patterns within the three study sites.

STARFM Data Fusion

The STARFM data fusion algorithm excelled at creating a complete Landsat resolution time series at the MODIS temporal scale. These time series also showcase that NDVI did not oversaturate due to the LAI range at these study sites. If saturation had been present, clipping would have existed in some or all of the peaks within the trend line. Clipping refers the flattening of wave peaks caused by values that exceed the intended range. Such results show that NDVI is an acceptable vegetation index to use when monitoring and measuring kudzu phenology changes.

Due to the potentially high LAI values of kudzu, 3.7 – 7.8 (Forseth and Innis 2004), the validity of NDVI as a sufficient method of kudzu monitoring should be compared with other VIs such as EVI. Like NDVI, EVI is based on the simple ratio with adjustments expressed in Equation 4.

$$EVI = G\frac{\rho(nir) - \rho(red)}{\rho(nir) + C_1\rho(red) - C_2\rho(blue) + L} \quad (4)$$

This is a more robust VI compared to NDVI because of the inclusion of the constants C_1 and C_2 along with the soil coefficient L (Jensen 2016). The two constants adjust for atmospheric scattering and absorption in the red and blue bands respectively. L is derived based on the type of soil underlying the vegetation and accounts for many of its reflection. These coefficients adjust for any background canopy noise associated with leaf litter, snow, etc (Huete et al. 2002).

Kudzu expands in a blanketing nature which partially or completely replaces any type of canopy that it consumes. The increased presence of chlorophyll increases the sensitivity of NDVI thus potentially producing over estimation of kudzu productivity (Huete et al. 2002). By comparing the NDVI time series produced in this study with that of EVI a conclusion can be made if these value are indeed accurate or have been inflated due to the phenologic and physiologic attributes of kudzu.

Combining "good" and unmasked Landsat-derived NDVI with filled and synthetic imagery aided in preserving the most Landsat data possible as seen in (Schmidt et al. 2015). All three image series represent the progression of the fusion process with (1) MODIS image, (2) Landsat image with mask applied, (3) synthetic Landsat-scale image

derived from the fusion process and (4) the final Landsat-scale image classified as either original, filled or synthetic.

The BFAST algorithm works best when presented with a complete set of data. This is because it is an additive decomposition model which functions by using the sum of the seasonal, linear and error components to express the observed trend (Verbesselt et al. 2010b; Schmidt et al. 2015). If there is a portion of the observed trend missing, and later inputted, then the additive trends will be skewed due to the lack of data.

Reliance on Landsat data alone would have produced a time series that exhibits numerous points of missing data caused by its unreliable 16-day temporal resolution as displayed in Figures 10 – 12. Interpolation between good pixels and images to produce a complete time series could have potentially been misleading based on the number of missing pixels. Incorporating MODIS data with the STARFM algorithm provided good pixels correlating with missing Landsat pixels to produce synthetic imagery.

With this brings uncertainty due to the spatial resolution of the MODIS pixels in relation to the area of each study site. As none of the sites were large enough to contain a complete MODIS pixel, the synthetic Landsat-scale pixels within the study sites were derived from mixed MODIS pixels containing spectral signatures from surrounding vegetation. Gao et al. (2006) does mention that the accuracy of detecting phenology changes using STARFM is dependent on the size of the study area. Locating pure coarse resolution pixels representing the land cover in question increases the accuracy of the synthetic data for similar but smaller areas. Without a land cover classification it is not clear if a pure kudzu pixel was used in the STARFM algorithm. Future studies could

incorporate a land cover classification to clarify this questions and if the absence of a pure kudzu MODIS scale pixel effects the results of this study.

BFAST Analysis

Results from the BFAST analyses substantiate the use of the STARFM fusion algorithm to produce a much more robust image collection compared to one that is composed only of Landsat or MODIS data. Analysis of these plots show a distinct difference in the phenologic dynamics observed at 250 m compared to those made at 30 m spatial resolutions. The increase in the number of breaks produced from the 30 m filled dataset shows that finer resolution is better at detecting phenologic change at the local level. Each break calculated by the BFAST algorithm was matched with the corresponding image, or band, in the filled image collection. This was possible because each image correlates to a MOD13Q1 composite.

Site #1 did not present as many breaks as was recorded for Sites #2 and #3. A number of approaches were made to determine the cause of these breaks but a concrete answer was not reached. Perry and Breathitt counties are both host to numerous wildfires every year which range in size from 10 acres to over 500 acres according to the Kentucky State Department of Forestry (M. Harp and R. Boggs, personal communication, 21 March 2016). A firsthand account offered by the KYTC District 10 (D. R. Gumm, personal communication, 16 March 2016) was that at least one fire in early 2014 disturbed part of the Kudzu infestation at Site #1. At this same time the KYTC, in collaboration with a team from the University of Kentucky, began studying the western section of this site by

sectioning it off into 30 ft. x 30 ft. plots to test chemical and mechanical methods of kudzu eradication.

The 2014 fire hampered their project because it allowed a patch of ragweed to take hold. Unfortunately these disturbances were not able to be detected as a statistically significant break most likely due to its position near the end of the time series not allowing enough data to accurately predict breaks. Our inability to confirm when kudzu was established also prohibits the detections of a kudzu signal. A finer spatial resolution image collection could also be used to better match that of the study plots to reduce the presence of mixed pixels especially for experimental control methods.

According to the Kentucky Wildland Forest Management fall fire season ranges from October to December correlating with deciduous species dropping their leaves. This also happens to the be the time in which both Sites #2 and #3 exhibited overlapping breaks in 2005 and 2007. The distance separating the two sites along KY Highway 1096 is approximately 1.5 miles making it possible that both could have been impacted by the same fire events. Site #1 is 30 miles northwest, possibly isolating it from these fire events and explain the lack of breaks during this period. A lack of consistent recording of each fire and of burn scar extent made this data too inconsistent to definitively know if a fire consumed any portion of these two sites.

Fires in eastern Kentucky are most often caused by arsonists and human carelessness and is exacerbated by periods of sustained, reduced rainfall and high temperatures (Maingi and Henry 2007). Changes in the weather can also affect how efficiently kudzu grows. As mention earlier this plant prefers temperatures ranging from 25°C and 30°C and precipitation of at least 100 cm per year (Forseth and Innis 2004; Lindgren et al.

2013). Future studies could potentially correlate temperature and precipitation fluctuations with trends and breaks detected using the BFAST analysis. For example, years with early freeze dates or sustained drought may provide reasoning as to why a low estimated NDVI value for kudzu was observed. Including burn scar maps using the methods described by Maingi (2005) using Landsat ETM+ data in this area of the state may also prove to be useful by providing spatial and temporal clues as to the cause of vegetation disturbance observed by breaks in the linear trend.

Limitations

A number of limitations were considered when attempting to explain why the methods used in this study did not adequately detect changes to the extent and intensity of kudzu infestations. Landsat and MODIS are both multispectral sensors that are calibrated to detect fewer than 20 bands within the electromagnetic spectrum. Hyperspectral sensors collect 100s of bands that are narrower and have the ability to accurately assign specific spectral signatures to individual plant species, (He et al. 2011). The approach taken in this study is limited by assessing only the visible red and NIR bands of the electromagnetic spectrum via the NDVI which was assumed to present a unique spectral signature for kudzu compared to surrounding vegetation. The results do not provide concrete evidence that this is a sound method and that finer spectral resolution combined with a fine temporal resolution may be necessary for detection.

A previous study that was successful in detecting kudzu utilized the hyperspectral airborne AVIRIS sensor, (Cheng, Tom, and Ustin 2007), with a spatial resolution of 4 m. The combination of higher dimensionality within the visible red and NIR bands, along

with the spatial resolution, could produce a more sensitive NDVI-like estimate better tailored to kudzu. This method could be employed in conjunction with the Illinois Department of Resources (J. Shimp, personal communication, 30 September 2015) technique of monitoring kudzu after the first freeze when is browns. This approach has the potential to be problematic because the sensor collects data only when specified and at a significant cost, and not on a continual basis. Long term time series could not be assembled using this data because of the lack of temporal continuity. Attaching a sensor to a UAV could potentially reduce these limitations but would require someone trained in using the machinery and software.

This study is also limited by the absence of field-collected reference data and collection of *in situ* data related to site specific kudzu growth, presence, absence, and disturbances. This reduces the ability to explain the results from the time series and BFAST analyses. Similar approaches involving other vegetation types include the measurement of cheatgrass extent detected with the use a combination of land cover classification in the Great Basin and field observations made at over 600 locations in the study area (Bradley and Mustard 2005). This method models a more in depth approach to monitoring this invasive species because it creates a detailed land cover classification to validate remotely sensed spectral signatures of vegetation types.

Creating a land cover classification for the area within the 5000 m buffered area could also be beneficial as suggested by the results from the simple spectral. This is because of the presences of slight overall variations in individual spectral profiles present between vegetation types. These results combined with those from the BFAST analysis do show that NDVI is a valid indicator of vegetation phenology change overtime due to the slight

variations between vegetation and kudzu. Referring back to Figure 12, kudzu and forest exhibit similar patterns in estimated annual NDVI values but forest cover tends to show more extreme values at both peaks and troughs. This signals suggests that this variation is a characteristics that could be used to distinguish the two species and verified with a land cover classification map.

CHAPTER 6: CONCLUSION

This research was conducted to explore the efficacy of synthetic imagery as a method for measuring and monitoring kudzu infestations in eastern Kentucky. The position of Kentucky along the northern edge of the North American kudzu extent makes this a study area ripe for understanding the phenological characteristics that allow kudzu to propagate so aggressively. With the use of data acquisition scripts written for GEE this research was able to acquire and process high dimensional Landsat and MODIS image collections. By fusing these two image collections together via STARFM a multiband image was produced containing the Landsat 30 m spatial resolution and MODIS 16-day temporal resolution.

Results showed that the finer spatial resolution of the synthetic data was better at detecting within site disturbances in the vegetation dynamics compared to using MODIS data alone. Sites #2 and #3 both exhibited 2-3 breaks for each of the six pixels examined from within each site. At the scale of these kudzu infestations MODIS data is too spatially coarse which potentially hides kudzu vegetation dynamics within spectral signatures from neighboring vegetation. MODIS pixels may not show similar breaks because of mixing in the coarser resolution pixels.

Adding a field component may have significantly increased the chance of firmly addressing the phenologic trend of these kudzu infestations. The outcome did produce a set of script that could be reused in future studies to handle big data for processing time

series decomposition of kudzu and other vegetation. As time goes on and the temporal dimensionality extends for Landsat and MODIS which will further increase the need for a robust time series decomposition method as presented here.

In situ data collection would aid in the validation of the STARFM and BFAST results, especially as it relates to invasive species monitoring. In order to collect the most effective data the collection process might follow the guidelines posed by EDDMapS which provides collectors with required data fields that must be completed before submitting any observation to their open source site. These standards are taken from the North American Weed Management Association's (NAWMA) Invasive Plant Mapping Standards Following these standards during the data collection process will ensure the comparison to existing data on other kudzu infestations. Table 8

Table **8** lists NAWMA's standards that are required when collecting data.

Data Collection Standards		
Reporter	Canopy Closure	Datum
Date Entered	Latitude	Ownership
Pest	Longitude	Location Description
Observation Date	Infested Area Size (sq.m)	Images
State and County		

Table 8. EDDMapS and NAWMA Data Collection Standards

Refinement of this study with the addition of field work, more refined approach to BFAST pixel selection within the buffered areas and possibly different image collections results might aid land managers when detecting and formulating kudzu management plans. The process developed in this study showcases the importance of big data

processing, such as Google Earth Engine, for acquiring and processing large amounts of raster-based and remotely sensed data. It highlights the efficiency of allowing massive computer networks to carry out these steps on the fly at a global scale using highly dimensional datasets.

This study also illustrates how the STARFM algorithm is used to create a derived continuous Landsat-scale time series of estimated NDVI for the use in monitoring kudzu and other types of vegetation phenology dynamics. Although this technique combined with the BFAST analysis is highly technical it would be improved upon with the addition of field-based research. Remote sensing and other geospatial techniques do make research of local and global phenomenon accessible away from the source but there is still a need to validate such results with field data. By combining both data collecting techniques a more complete picture of kudzu phenologic dynamics can be made which will make monitoring and eradication for efficient. Understanding how to use and interpret remotely sensed data requires substantial training which does limit its use and would require collaboration on mitigation efforts if needed. Kudzu will continue to threaten landscapes, economies and communities and this is another step towards a solution to the problem.

REFERENCES

Asner, G. P., D. E. Knapp, T. Kennedy-Bowdoin, M. O. Jones, R. E. Martin, J. Boardman, and R. F. Hughes. 2008. Invasive species detection in Hawaiian rainforests using airborne imaging spectroscopy and LiDAR. *Remote Sensing of Environment* 112 (5):1942–1955.

Bender, J. personal communication. 18 Spetember 2015.

Blaustein, R. J. 2001. Kudzu's invasion into Southern United States life and culture. *The Great Reshuffling: Human Dimensions of Invasive Species.* 55–62.

Birth, G. S., and G. R. McVey. 1968. Measuring the color of growing turf with a reflectance spectrophotometer. *Agronomy Journal.* 60 (6): 640-643.

Bradley, B. A., and J. F. Mustard. 2005. Identifying land cover variability distinct from land cover change: Cheatgrass in the Great Basin. *Remote Sensing of Environment.* 94 (2):204–213.

Buheaosier, K., Tsuchiya, M. K. and S. J. S. 2003. Comparison of Image Data Acquired with AVHRR, MODIS, ETM+ and ASTER over Hokkaido, Japan. *Advances in Space Research*. 32 (11):2211–2216.

Callen, S. T., and A. J. Miller. 2015. Signatures of niche conservatism and niche shift in the North American kudzu (*Pueraria montana*) invasion. *Diversity and Distributions.* 21 (8):853–863.

Cheng, Y. B., E. Tom, and S. L. Ustin. 2007. Mapping an invasive species, kudzu (*Pueraria montana*), using hyperspectral imagery in western Georgia. *Journal of Applied Remote Sensing*. 1 (1):013514.

Chuvieco, E., S. Opazo, W. Sione, H. Valle, J. Anaya, D. Bella, I. Cruz, L. Manzo, G. López, N. Mari, F. González-alonso, F. Morelli, A. Setzer, I. Csiszar, J. A. Kanpandegi, A. Bastarrika, R. Libonati, S. E. Applications, and N. Jan. 2013. Global Burned-Land Estimation in Latin America Using Modis Composite Data. *Ecological Applications.* 18 (1):64–79.

EDDMapS. 2016. Early Detection & Distribution Mapping System. The University of Georgia - Center for Invasive Species and Ecosystem Health. Available online at http://www.eddmaps.org/; (last accessed 20 March 2016).

ESRI. 2014. ArcGIS Desktop v10.3.1. Redlands, CA.: Environmental Systems Research Institute.

Forkel, M., N. Carvalhais, J. Verbesselt, M. D. Mahecha, C. S. R. Neigh, and M. Reichstein. 2013. Trend Change detection in NDVI time series: Effects of inter-annual variability and methodology. *Remote Sensing.* 5 (5):2113–2144.

Forseth, I. N., and A. F. Innis. 2004. Kudzu (*Pueraria montana*): History, Physiology, and Ecology Combine to Make a Major Ecosystem Threat. *Critical Reviews in Plant Sciences.* 23 (5):401–413.

Gao, F., J. Masek, M. Schwaller, and F. Hall. 2006. On the blending of the landsat and MODIS surface reflectance: Predicting daily landsat surface reflectance. *IEEE Transactions on Geoscience and Remote Sensing.* 44 (8):2207–2218

Google EarthPro. Google

Goward, S., T. Arvldson, D. Williams, J. Faundeen, J. Irons, and S. Franks. 2006. Historical record of landsat global coverage : Mission operations, NSLRSDA, and international cooperator stations. *Photogrammetric engineering and remote sensing* 72 (10):1155–1169.

Gu, Y., and B.K. Wylie. Develeoping a 30-m grassland productivity estimation map for central Nebraska using 250-m MODIS and 30-m Landsat-8 observations. *Remote Sensing of Environment.* 171:291-298.

Gumm, D. R., personal communication. 29 October 2015.

Gumm, D. R., personal communication. 16 March 2016.

Hansen, M. C., P. V Potapov, R. Moore, M. Hancher, S. A. Turubanova, and A. Tyukavina. 2013. High-Resolution Global Maps of 21st Century Forest Cover Change. *Science.* 342 (6160):850–853.

Harp, M. and R. Boggs. personal communication. 21 March 2016.

Hawthorne, T. L., V. Elmore, A. Strong, P. Bennett-Martin, J. Finnie, J. Parkman, T. Harris, J. Singh, L. Edwards, and J. Reed. 2015. Mapping non-native invasive species and accessibility in an urban forest: A case study of participatory mapping and citizen science in Atlanta, Georgia. *Applied Geography.* 56:187–198.

He, K. S., D. Rocchini, M. Neteler, and H. Nagendra. 2011. Benefits of hyperspectral remote sensing for tracking plant invasions. *Diversity and Distributions.* 17 (3):381–392.

Hilker, T., M. A. Wulder, N. C. Coops, J. Linke, G. McDermid, J. G. Masek, F. Gao, and J. C. White. 2009. A new data fusion model for high spatial- and temporal-resolution mapping of forest disturbance based on Landsat and MODIS. *Remote Sensing of Environment.* 113 (8):1613–1627.

Hill, J and H. M. Mogil., 2012. The Weather and Climate of Kentucky. *Weatherwise.* 65 (5):26–32.

Huang, C., and G. P. Asner. 2009. Applications of Remote Sensing to Alien Invasive Plant Studies. *Sensors.* 9 (6):4869–4889.

Huete, A. R., H. Q. Liu, K. Batchily, and W. Van Leeuwen. 1997. A comparison of vegetation indices over a global set of TM images for EOS-MODIS. *Remote Sensing of Environment.* 59 (3):440–451.

Huete, A., K. Didan, T. Miura, E. P. Rodriguez, X. Gao, and L. G. Ferreira. 2002. Overview of the radiometric and biophysical performance of the MODIS vegetation indices. *Remote Sensing of Environment.* 83 (1-2):195–213.

Hunt, R., R. Hamilton, and J. Everitt. n.d. What Weeds Can Be Remotely Sensed?. *United States Department of Agriculture: United States Forest Service.* 1 – 5.

Hutchinson, J. M. S., A. Jacquin, S. L. Hutchinson, and J. Verbesselt. 2015. Monitoring vegetation change and dynamics on U.S. Army training lands using satellite image time series analysis. *Journal of Environmental Management.* 150:355–366.

Irons, J. R., J. L. Dwyer, and J. A. Barsi. 2012. The next Landsat satellite: The Landsat Data Continuity Mission. *Remote Sensing of Environment.* 122:11–21.

Jakubauskas, M. E., D. R. Legates, and J. H. Kastens. 2001. Harmonic analysis of time - series AVHRR NDVI data. *Photogrammetric Engineering and Remote Sensing.* 67 (4):461 – 470.

Jensen, J. R. 2016. *Introductory Digital Image Processing: A Remote Sensing Perspective*. Pearson Education Inc.

Jin, H., and L. Eklundh. 2014. A physically based vegetation index for improved monitoring of plant phenology. *Remote Sensing of Environment.* 152:512–525.

Johnson, L. F., D. E. Roczen, S. K. Youkhana, R. R. Nemani, and D. F. Bosch. 2003. Mapping vineyard leaf area with multispectral satellite imagery. *Computers and Electronics in Agriculture.* 38 (1):33–44.

Lhermitte, S., J. Verbesselt, W. W. Verstraeten, and P. Coppin. 2011. A comparison of time series similarity measures for classification and change detection of ecosystem dynamics. *Remote Sensing of Environment.* 115 (12):3129–3152.

Lindgren, C. J., K. L. Castro, H. A. Coiner, R. E. Nurse, and S. J. Darbyshire. 2013. The Biology of Invasive Alien Plants in Canada. *Pueraria montana* var. *lobata* (Willd.). *Canadian Journal of Plant Science.* 93 (1):71–95.

Lobell, D.B., D, Thau, C. Seifert, E. Engle, and B. Little. A scalable satellite-based crop yield mapper. *Remote Sensing of Environment*. 164: 324-333.

LP DAAC. 2014. Vegetation Indices 16-Day L3 Global 250m. https://lpdaac.usgs.gov/dataset_discovery/modis/modis_products_table/mod13q1. (last accessed 5 April 2016).

Maiersperger, T. K., P. L. Scaramuzza, L. Leigh, S. Shrestha, K. P. Gallo, C. B. Jenkerson, and J. L. Dwyer. 2013. Characterizing LEDAPS surface reflectance products by comparisons with AERONET, field spectrometer, and MODIS data. *Remote Sensing of Environment.* 136:1–13.

Maingi, J. K. 2005. Mapping Fire Scars in a Mixed-Oak Forest in Eastern Kentucky, USA, Using Landsat ETM+ Data. *Geocarto International* 20 (3):51–63.

Maingi, J. K., and M. C. Henry. 2007. Factors influencing wildfire occurrence and distribution in eastern Kentucky, USA. *International Journal of Wildland Fire.* 16 (1):23–33.

Maselli, F. 2004. Monitoring forest conditions in a protected Mediterranean coastal area by the analysis of multiyear NDVI data. *Remote Sensing of Environment.* 89 (4):423–433.

Muchoney, D., J. Borak, H. Chi, M. Friedl, S. Gopal, J. Hodges, N. Morrow, and A. Strahler. 2000. Application of the MODIS global supervised classification model to vegetation and land cover mapping of Central America. *International Journal of Remote Sensing.* 21 (6-7):1115–1138.

Padarian, J., B. Minasny, and A. B. McBratney. 2015. Using Google's cloud-based platform for digital soil mapping. *Computers & Geosciences.* 83:80–88.

Patel, N. N., E. Angiuli, P. Gamba, A. Gaughan, G. Lisnini, F. R. Stevens, A. J. Tatem, and G. Trianni. 2014. Multitemporal settlement and population mapping from Landsat using Google Earth Engine. *International Journal of Applied Earth Observation and Geoinformation.* 35: 199-208.

R Core Team. 2016. R: A Language and Environment for Statistical Computing. http://www.r-project.org/ (last accessed 4 April 2016).

Rafique, R., F. Zhao, R. de Jong, N. Zeng, and G. Asrar. 2016. Global and Regional Variability and Change in Terrestrial Ecosystems Net Primary Production and NDVI: A Model-Data Comparison. *Remote Sensing* 8 (3):177.

Rouse Jr, J.W., R. H. Haas, J. A. Schell, and D. W. Deering. Monitoring vegetation systems in the Great Plains with ERTS. *NASA special publication.* 351 (1974): 309.

Salajanu, D., and D. M. Jacobs. 2009. Using Forest Inventory Plot Data and Satellite Imagery From Modis and Landsat-Tm To Model Spatial Distribution Patterns of Honeysuckle and Privet. In *USDA Forest Service*.

Schmidt, M., R. Lucas, P. Bunting, J. Verbesselt, and J. Armston. 2015. Multi-resolution time series imagery for forest disturbance and regrowth monitoring in Queensland, Australia. *Remote Sensing of Environment.* 158:156–168.

Shimp, J. personal communication. 30 September 2015.

Shouse, M., L. Liang, and S. Fei. 2013. Identification of understory invasive exotic plants with remote sensing in urban forests. *International Journal of Applied Earth Observation and Geoinformation* 21:525–534.

Silleos, N. G., T. K. Alexandridis, I. Z. Gitas, and K. Perakis. 2006. Vegetation Indices: Advances Made in Biomass Estimation and Vegetation Monitoring in the Last 30 Years. *Geocarto International.* 21 (4):21–28.

Smith, C. 2010. Invasive Exotic Plants of North Carolina: Pueraria montana (Kudzu). *N.C. Department of Transportation. Raleigh, NC.* (08):1–2.

Turner, D. P., W. B. Cohen, R. E. Kennedy, K. S. Fassnacht, and J. M. Briggs. 1999. Relationships between leaf area index and Landsat TM spectral vegetation indices across three temperate zone sites. *Remote Sensing of Environment.* 70 (1):52–68.

United States Department of Agriculture. 2016. STARFM. *Software Download*. http://www.ars.usda.gov/services/software/download.htm?softwareid=432. (last accessed 5 April 2016).

USGS. 2016. PRODUCT GUIDE LANDSAT 4-7 CLIMATE DATA RECORD (CDR) SURFACE REFLECTANCE. *Version 6.3* (March). http://landsat.usgs.gov/documents/ledaps_release_notes.pdf (last accessed 4 April 2016).

Verbesselt, J., R. Hyndman, G. Newnham, and D. Culvenor. 2010a. Detecting trend and seasonal changes in satellite images time series. *Remote Sensing of Environment* (114):106–115.

Verbesselt, J., R. Hyndman, A. Zeileis, and D. Culvenor. 2010b. Phenological change detection while accounting for abrupt and gradual trends in satellite image time series. *Remote Sensing of Environment* 114 (12):2970–2980.

Walker, J. J., K. M. De Beurs, R. H. Wynne, and F. Gao. 2012. Evaluation of Landsat and MODIS data fusion products for analysis of dryland forest phenology. *Remote Sensing of Environment.* 117:381–393.

Wang, P., F. Gao, and J. G. Masek. 2014. Operational data fusion framework for building frequent landsat-like imagery. *IEEE Transactions on Geoscience and Remote Sensing.* 52 (11):7353–7365.

Wulder, M. A., J. G. Masek, W. B. Cohen, T. R. Loveland, and C. E. Woodcock. 2012. Opening the archive: How free data has enabled the science and monitoring promise of Landsat. *Remote Sensing of Environment* 122:2–10.

Yang, D. 2012. MODIS-Landsat Data Fusion for Estimating Vegetation Dynamics–A Case Study for Two Ranches in West Texas. *10th Annual TAMUS Pathways Student Research Symposium*.

Zhang, X., M. A. Friedl, C. B. Schaaf, and A. H. Strahler. 2004. Climate controls on vegetation phenological patterns in northern mid- and high latitudes inferred from MODIS data. *Global Change Biology.* 10 (7):1133–1145.

Zheng, G. and L.M. Moskal. 2009. Retrieving Leaf Area Index (LAI) Using Remote Sensing: Theories, Methods and Sensors. *Sensors*. 9(4):2719-2745.

Zhitao, Z., Y. Lan, W. Pute, and H. Wenting. 2014. Model of soybean NDVI change based on time series. *International Journal of Agriculture and Biological Engineering.* 7 (5):64–70.

APPENDIX A: GOOGLE EARTH CODE

```
var geometry = /* color: 98ff00 */ee.Geometry.Point([-
83.54278564453125, 37.53477698849114]);
var modis = ee.ImageCollection("MODIS/MOD13Q1");
var subset_feature =
ee.FeatureCollection("ft:1FxoaelgbPUVzVl4rOcrZCnvN3ZE18ZlWtv1IaLZ
p").aside(print);

////  Landsat 32 day NDVI composites for Landsat TM, ETM+ and
OLI:
var landsat5 = ee.ImageCollection('LANDSAT/LT5_SR'); // Jan 1,
1984 - May 5, 2012
var landsat7 = ee.ImageCollection('LANDSAT/LE7_SR'); // Jan 1,
1999 - Feb 14, 2016
var landsat8 = ee.ImageCollection('LANDSAT/LC8_SR'); // Apr 11,
2013 - Nov 1, 2015

////  MODIS 16-day VI composites (start in March so we have a
month
////    to composite with before hand for Landsat):
var filtered_modis = modis.filterDate('2000-02-01', '2015-12-
31');

////  Both image collections are filtered_ls for the dates that
are
////    in question (a supplemental period in 2007 is required)
////    from Landsat 7 to fill a gap in Landsat 5 data), see
below:
var filtered_ls5 = landsat5.filterDate('1999-12-01', '2015-12-
31');
var filtered_ls7 = landsat7.filterDate('1999-12-01', '2015-12-
31');
var filtered_ls8 = landsat8.filterDate('1999-12-01', '2015-12-
31');


////  Extract feature collection bounds for clipping and region
exporting:
////    This sets the correct correct coordinte system and
spatial
////        resolution...
var subset_bounds =
subset_feature.geometry().transform('EPSG:4326',
30).bounds().getInfo();


////  Function to remove LEDAPS snowy and cloudy pixels based on
QA band, but retain
////    system time:
```

```
49
50  var removeBadObservations = function(image){
51    var valid_data_mask = ee.Image(image).select('cfmask').lte(1);
52
53    var numberBandsHaveData =
54  image.mask().reduce(ee.Reducer.sum());
55    var allOrNoBandsHaveData =
56  numberBandsHaveData.eq(0).or(numberBandsHaveData.gte(9));
57    var allBandsHaveData = allOrNoBandsHaveData;
58
59    //Make sure no band is just under zero
60    var allBandsGT = image.reduce(ee.Reducer.min()).gt(-0.001)
61    var result =
62  ee.Image(image).mask(image.mask().and(valid_data_mask).and(allBan
63  dsHaveData).and(allBandsGT));
64    return
65  result.copyProperties(ee.Image(image),['system:time_start']);
66  };
67
68  ////  Functions to calculate NDVI for different sensors:
69  var getNDVI_tm = function(image){
70      var ndvi = ee.Image(image).normalizedDifference(['B4','B3']);
71      return
72  ndvi.copyProperties(ee.Image(image),['system:time_start']);
73  };
74  var getNDVI_oli = function(image){
75      var ndvi = ee.Image(image).normalizedDifference(['B5','B4']);
76      return
77  ndvi.copyProperties(ee.Image(image),['system:time_start']);
78  };
79
80
81  ////  Filter the filtered_ls collecitons by the bound, remove bad
82  pixels,
83  ////    and calculate NDVI for all remaining images:
84  filtered_ls5 =
85  filtered_ls5.filterBounds(subset_bounds).aside(print).map(removeB
86  adObservations).map(getNDVI_tm);
87  filtered_ls7 =
88  filtered_ls7.filterBounds(subset_bounds).aside(print).map(removeB
89  adObservations).map(getNDVI_tm);
90  filtered_ls8 =
91  filtered_ls8.filterBounds(subset_bounds).aside(print).map(removeB
92  adObservations).map(getNDVI_oli);
93
94  ////  Combine image collections across sensors:
95  var filtered_ls_ndvi = filtered_ls5;
96  ////  Add only select ranges of ETM+ data:
97  filtered_ls_ndvi =
98  filtered_ls_ndvi.merge(filtered_ls7.filterDate('2007-10-01',
99  '2007-12-31'));
100 filtered_ls_ndvi =
101 filtered_ls_ndvi.merge(filtered_ls7.filterDate('2011-04-01',
102 '2013-06-30'));
103 filtered_ls_ndvi = filtered_ls_ndvi.merge(filtered_ls8);
104
105
```

```
////  Convert polygon boundaries to raster by using the paint()
////    function on a non-existent property, and then add 1:
var subset_mask = ee.Image().byte().paint(subset_feature,
"id").add(1);


////  Process MODIS data to time series:
filtered_modis =
filtered_modis.filterBounds(subset_bounds).aside(print);

var extract_modis_date = function(row) {
  ////  Pull out the date:
  var d = ee.Date(row.get('system:time_start'));
  var d2 = ee.Date.fromYMD(d.get('year'), d.get('month'),
d.get('day'));
  var result = ee.Feature(null, {'date': d2});
  result = result.set({'date': d2});
  return result;
};

/*
 * A function that returns an image containing just the specified
QA bits.
 *
 * Args:
 *   image - The QA Image to get bits from.
 *   start - The first bit position, 0-based.
 *   end   - The last bit position, inclusive.
 *   name  - A name for the output image.
 */
var getQABits = function(image, start, end, newName) {
  ////  Compute the bits we need to extract.
  var pattern = 0;
  for (var i = start; i <= end; i++) {
     pattern += Math.pow(2, i);
  }
  ////  Return a single band image of the extracted QA bits,
giving the band
  ////    a new name.
  return image.select([0], [newName])
                .bitwiseAnd(pattern)
                .rightShift(start);
};

////  Process MODIS data, first subsetting and masking:
filtered_modis = filtered_modis.map(function(image){
  ////  Check to see if the detailed QA bits 4-5 are both 11,
indicating
  ////    bottom level of "Decreasing quality..." and set it to 1
if not
  ////    and zero if so to use in masking MODIS data:
  ////
https://lpdaac.usgs.gov/dataset discovery/modis/modis products ta
ble/mod13q1
  var quality = getQABits(image.select(2), 4, 5, 'QAMask');
  quality = quality.eq(3).not();
```

```
  return
image.clip(subset_bounds).mask(image.mask().multiply(subset_mask)
.multiply(quality));
});

////  We also have to select our Day of Year and NDVI:
var filtered_modis_day = filtered_modis.select(10);
filtered_modis = filtered_modis.select(0);


////  Construct a multiband image from the image collection:
var modis_multiband = filtered_modis.filterDate('2000-03-01',
'2015-12-31').iterate( function(x, modis_multiband) {
  return ee.Image(modis_multiband).addBands(ee.Image(x));
}, filtered_modis.first());
var modis_day_multiband = filtered_modis_day.filterDate('2000-03-
01', '2015-12-31').iterate( function(x, modis_day_multiband) {
  return ee.Image(modis_day_multiband).addBands(ee.Image(x));
}, filtered_modis_day.first());


////  Construct date set from our MODIS image collection:
var dates_modis = filtered_modis.map(extract_modis_date);
print(dates_modis.getInfo());
////  Not needed to export because we export MODIS/Landsat
together
////    below:
//Export.table(dates_modis, 'Subset_MODIS_NDVI_16day_Dates');


////  Process Landsat across MODIS dates:

////  Apply a subset and mask:
filtered_ls_ndvi = filtered_ls_ndvi.map(function(image) {
  return ee.Image(image)
    .clip(subset_bounds)
    .mask(
      ee.Image(image)
      .mask()
      .multiply(subset_mask));
});


////  Reduce the collection to a new collection by dates,
averaging all
////    observations across three Landsat observations (the first
will only
////    two months):

////  Construct a potential composite of +/- X days. Choose 15 if
////    you only want the possibility of one Landsat scene per
////    MODIS image date:
var day_expand = 16;

var reduceLandsatNDVI = function(MODISdate) {
  MODISdate = ee.Date(MODISdate.get('date'));

```

```
219     ////  The MODIS time_start represents the beginning of the
220   MODIS composite
221     ////    window.  Therefore, to extract a Landsat scene that was
222   captured
223     ////    within the same composite period, we will not look
224   before and after
225     ////    that date:
226     //var ndvi_subset =
227   ee.ImageCollection(filtered_ls_ndvi).filterDate(
228   MODISdate.advance(-1*day_expand, 'day'),
229   MODISdate.advance(day_expand, 'day') );
230     ////  But instead look only after that date the width of the
231   day_expand
232     ////    which represents the composite window:
233     var ndvi_subset =
234   ee.ImageCollection(filtered_ls_ndvi).filterDate( MODISdate,
235   MODISdate.advance(day_expand, 'day') );
236
237     ////  Calculate absolute value difference from target date,
238     ////    this will find the Landsat image nearest the MODIS
239   date:
240     ndvi_subset = ndvi_subset.map(function (image) {
241       var diff =
242   MODISdate.difference(ee.Date(ee.Image(image).get('system:time_sta
243   rt')), 'day').abs();
244       return ee.Image(image).set('diff', diff);
245     });
246
247     ndvi_subset = ndvi_subset.sort('diff');
248     var ndvi_first = ndvi_subset.reduce('first');
249     var ndvi_mean = ndvi_subset.reduce('mean');
250
251     ////  Anywhere this is zero, calculate the mean across the
252   collection
253     ////    as an alternative for the missing/bad data.  Note the
254   use
255     ////    of the mask() to undo the calculation masking for areas
256     ////    previously excluded due to clouds, etc.:
257     return ee.Algorithms.If(
258       ndvi_first.bandNames(),
259       ndvi_first.eq(0).multiply(ndvi_mean).add(ndvi_first),
260
261       // Workaround reduceRegion() failing on images
262       // with no bands.
263       ee.Image(0)
264     );
265   };
266
267   var extract_landsat_date = function(MODISdate) {
268     MODISdate = ee.Date(MODISdate.get('date'));
269
270     ////  See note above about the composite window and looking
271   forward
272     ////    from the MODIS time_start:
273     //var ndvi_subset =
274   ee.ImageCollection(filtered_ls_ndvi).filterDate(
```

```
MODISdate.advance(-1*day_expand, 'day'),
MODISdate.advance(day_expand, 'day') );
  var ndvi_subset =
ee.ImageCollection(filtered_ls_ndvi).filterDate( MODISdate,
MODISdate.advance(day_expand, 'day') );

  ////  Calculate absolute value difference from target date,
  ////    this will find the Landsat image nearest the MODIS
date:
  ndvi_subset = ndvi_subset.map(function (image) {
    var diff =
MODISdate.difference(ee.Date(ee.Image(image).get('system:time_sta
rt')), 'day').abs();
    return ee.Image(image).set('diff', diff);
  });

  ndvi_subset = ndvi_subset.sort('diff');
  var d = ndvi_subset.aggregate_first('system:time_start');
  var count = ndvi_subset.aggregate_count('system:time_start');

  ////  Pull out the date:
  d = ee.Algorithms.If(
    ee.Number(count).gt(0),
    ee.Date(d),
    ee.Date('1971-01-01')
  );
  d = ee.Date(d);

  var d2 = ee.Date.fromYMD(d.get('year'), d.get('month'),
d.get('day'));

  var result = ee.Feature(null, {'LSdate': d2, 'MODISdate':
MODISdate, 'CountLSScenes': count});
  result = result.set({'LSdate': d2, 'MODISdate': MODISdate,
'CountLSScenes': count});
  return result;
};

var ls_collection = dates_modis.map(reduceLandsatNDVI);

////  Construct date set from our MODIS image collection:
var dates_landsat = dates_modis.map(extract_landsat_date);
//print(dates_landsat.getInfo());
Export.table(dates_landsat,
'South_3_Subset_Matching_NDVI_16day_Dates');


////  Construct a multi-band image from the image collection,
////    stripping the first item from the collection so we don't
////    duplicate it:
var ls_multiband = ls_collection.filterMetadata('system:index',
'not_equals', 'MOD13Q1_005_2000_03_05').iterate( function(x,
ls_multiband) {
  return ee.Image(ls_multiband).addBands(ee.Image(x));
}, ls_collection.first());

////  Scale the -1 to 1 values by 10000 and convert to integer
```

```
////    to match MODIS, then convert zeroes to missing (by-
product
////    of ND calculation and missing reducer values):
ls_multiband = ee.Image(ls_multiband).multiply(10000).int16();
ls_multiband =
ls_multiband.mask(ls_multiband.mask().multiply(ls_multiband.neq(0
)));


////  The multiband stack should now contain layers for each
month
////    including the supplemental ones pulled from each sensor
stack:
print(modis_multiband);
print(ls_multiband);


////  NOTE: I'm exporting the MODIS at 30m as well to match the
Landsat
////    stack:
Export.image(modis_multiband, 'South_3_Subset_MODIS_NDVI_16day',
{
  crs:'EPSG:32617',
  region:subset_bounds,
  scale:30
});
Export.image(modis_day_multiband,
'South_3_Subset_MODIS_DoY_16day', {
  crs:'EPSG:32617',
  region:subset_bounds,
  scale:30
});
////  Exporting the image as a raster data set that can be opened
in ENVI
Export.image(ls_multiband, 'South_3_Subset_Landsat_NDVI_16day', {
  crs:'EPSG:32617',
  region:subset_bounds,
  scale:30
});


Map.addLayer(subset_feature, {color: 'FF0000'}, 'Subset');
//var coords =
ee.Feature(subset.first()).centroid().geometry().coordinates();
//print(ee.Number(coords.get(1)));
Map.centerObject(subset_feature.first(), 12);
//Map.setCenter(-83.54, 37.53, 12);

Map.addLayer(ee.Image(modis_multiband), {}, 'MODIS NDVI
Composites');
Map.addLayer(ee.Image(modis_day_multiband), {}, 'MODIS Day of
Year Composites');
Map.addLayer(ee.Image(ls_multiband), {}, 'Landsat NDVI
Composites');
```

APPENDIX B: R PROGRAMMING CODE FOR STARFM

```
##
folder <- "C:/tmp"

setwd(folder)

#install.packages("rgdal")
#install.packages("raster")
library(raster)

#install.packages("bfast")
library(bfast)

#install.packages("rgdal")
library(rgdal)

#install.packages("animation")
library(animation)

#install.packages("ggplot2")
library(ggplot2)


modis <- brick("./data/South_Site_3_Subset_MODIS_NDVI_16day.tif")
landsat <-
brick("./data/South_Site_3_Subset_Landsat_NDVI_16day.tif")
study_area <- readOGR("./data", "South_Site_3")
poi <- readOGR("./data", "South_Site_3_Point_Merge")


##  Read imagery dates and matching data:
image_dates <-
read.csv("./data/South_Site_3_Subset_Matching_NDVI_16day_Dates.cs
v", stringsAsFactors=F)


##  Plotting defaults:

##  Generate color ramp to use:
z.lim = c(-2500,10500)
r.brks <- seq(z.lim[1], z.lim[2], by=(z.lim[2]-z.lim[1])/254)
#color_vec <- colorRampPalette(c("navyblue", "steelblue",
"limegreen", "yellow", "#FEFEFE"))(255)
color_vec <- colorRampPalette(c("AntiqueWhite1",
"darkgreen"))(255)

## Placeholder raster for legend creation:
r.leg <- raster(nrow=10,ncol=10)
r.leg[] <- 0
```

```
49
50  ##  If data has already been processed, load the results:
51  modis <- brick("./output/Subset_MODIS_NDVI_16day_masked.envi")
52  landsat <-
53  brick("./output/Subset_Landsat_NDVI_16day_masked.envi")
54
55  landsat_sim <-
56  brick("./output/Subset_Landsat_NDVI_16day_sim.envi")
57  landsat_filled <-
58  brick("./output/Subset_Landsat_NDVI_16day_filled.envi")
59
60  ##########################################
61
62
63  ##  Set StarFM_config.txt to have dimensions that match. Note
64  ##      that all other settings are considered fixed and would
65  ##      need to be modified by hand (if data type or NA values
66  ##      vary, for example):
67  config <- readLines("./src/StarFM_config.txt")
68  config <- gsub("(.*NROWS = ).*$", paste0("\\1", nrow(landsat)),
69  config )
70  config <- gsub("(.*NCOLS = ).*$", paste0("\\1", ncol(landsat)),
71  config )
72  cat(config, file="./src/StarFM_config.txt", sep="\n")
73
74
75  ##  Simulate one intermediate data period to test the StarFM
76  operation:
77  #modis_t1 <- modis[[1]]
78  #modis_t2 <- modis[[2]]
79  #modis_t3 <- modis[[3]]
80  #landsat_t1 <- landsat[[1]]
81  #landsat_t3 <- landsat[[2]]
82  #
83  #writeRaster(modis_t1, filename="./tmp/modis_t1.envi",
84  bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
85  #writeRaster(modis_t2, filename="./tmp/modis_t2.envi",
86  bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
87  #writeRaster(modis_t3, filename="./tmp/modis_t3.envi",
88  bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
89  #writeRaster(landsat_t1, filename="./tmp/landsat_t1.envi",
90  bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
91  #writeRaster(landsat_t3, filename="./tmp/landsat_t3.envi",
92  bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
93  #
94  #system2(command="./lib/StarFM/source/StarFM.exe",
95  args="./src/StarFM_config.txt", wait=TRUE)
96
97
98  ##  Threshold for "good" non-missing data that represents a
99  ##    "matchable" Landsat scene (should be set to ~10% of non-
100 ##      masked pixels.  These non-masked pixels :
101 ##  To calculate, something like the following after finding a
102 layer with
103 ##      all good data:
104
```

```
##  Fix any missing data that enters as zeroes to our NA value.
We have
##      to do this by layer as operating on the entire stack may
run into
##      memory issues on larger subsets:
for (i in 1:nlayers(modis)) {
    ##  Use the raster Which() function for speed:
    masked <- Which(modis[[i]] == 0, cells=TRUE)
    modis[[ i ]][ masked ] <- -32768
    masked <- Which(landsat[[i]] == 0, cells=TRUE)
    landsat[[ i ]][ masked ] <- -32768
}
writeRaster(modis,
filename="./output/Subset_MODIS_NDVI_16day_masked.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
writeRaster(landsat,
filename="./output/Subset_Landsat_NDVI_16day_masked.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
##  Or load them if already compiled:
modis <- brick("./output/Subset_MODIS_NDVI_16day_masked.envi")
landsat <-
brick("./output/Subset_Landsat_NDVI_16day_masked.envi")


##  Find a good layer where missing data is almost zero except
for
##      edge/projection/export issues coming out of Google Earth
Engine:
good_layer <- 6
plot(landsat[[good_layer]])
min_bad <- 0.05
masked <- sum(is.na(landsat[[good_layer]][]))
pct_good_unmasked <- (min_bad*( ncell(landsat) - masked) +
masked) / ncell(landsat)


##  Construct good/missing data vector:
stats <- image_dates
stats[["LSdate"]] <- as.Date( substr(image_dates$LSdate, 1, 10) )
stats[["MODISdate"]] <- as.Date( substr(image_dates$MODISdate, 1,
10) )
##  Strip Null geometry field:
stats <- stats[,-5]

for (i in 1:nlayers(landsat)) {
    num_ls_bad <- sum(Which(is.na(landsat[[i]]))[])
    num_modis_bad <- sum(Which(is.na(modis[[i]]))[])
    stats[i, "LSbad"] <- num_ls_bad >=
pct_good_unmasked*ncell(landsat)
    stats[i, "MODISbad"] <- num_modis_bad >=
pct_good_unmasked*ncell(modis)
    stats[i, "LSbad_pct"] <- (num_ls_bad - masked) /
(ncell(landsat) - masked)
    stats[i, "MODISbad_pct"] <- (num_modis_bad - masked) /
(ncell(modis) - masked)
}
landsat_bad <- stats[["LSbad"]]
```

```
modis_bad <- stats[["MODISbad"]]


##  Save off compiled dates and stats for future use:
write.csv(stats, file="./output/Image_Date_Statistics.csv")


##  Construct time series out of LS and MODIS dates for plotting:
ls_plot <- stats[ stats$LSdate > as.Date("2000-01-01"), ]

##  Quick and dirty way to plot MODIS irregularity:
#plot(xts(stats_plot,
order.by=stats_plot$MODISdate)$MODISbad_pct, type="o")

##  Create a PDF plot of image dates from compiled MODIS and
Landsat scenes:
pdf(file="./output/image_dates.pdf", width=10, height=3)
plot(ls_plot$LSdate, 1 - ls_plot$LSbad_pct, main="MODIS and
Landsat Image Dates", col=ifelse(!ls_plot$LSbad, "#00990066",
"#99000066"), type="h", ylim=c(0,1.5), ylab="Good Pixel
Proportion", xlab="Acquisition Date")
points(ls_plot$LSdate, (1- ls_plot$LSbad_pct),
col=ifelse(!ls_plot$LSbad, "#00990066", "#99000066"), pch=20,
cex=(1-as.numeric(ls_plot$LSbad_pct)))
points(stats$MODISdate, (1 - stats$MODISbad_pct),
col="#00004466", pch=3, cex=0.8)
legend(
    "topright",
    horiz=T,
    legend=c("Landsat Above 95%", "Landsat Below 95%", "MODIS
Scenes"),
    col=c("#00990066", "#99000066", "#00440066"),
    lty=c(1,1,0),
    pch=c(20, 20, 3),
    pt.cex=c(1, 1, 0.8),
    bg="white",
    cex=0.6
)
dev.off()



##  If the above all works, then we run the following to loop
over the
##      MODIS time steps, filling in Landsat output as we go:
landsat_sim <- stack(modis)
landsat_sim[] <- NA
landsat_filled <- stack(modis)
landsat_filled[] <- NA
##  Or load it if we already ran it:
#landsat_sim <-
brick("./output/Subset_Landsat_NDVI_16day_sim.envi")


##  Iterate and run StarFM for each MODIS date, choosing the
##      nearest pair of good MODIS/Landsat dates, one before and
##      one after the date being simulated where possible:
```

```
pb <- pbCreate((nlayers(landsat_sim)), "window", style=3,
label='Time Step Progress')
for (i in 1:nlayers(landsat_sim)) {
    ##  Determine next good Landsat scene in time series to
estimate from:
    ##      NOTE: The special case for estimating the first and
last scenes
    ##          of the series are handled:
    if (i == 1) {
        back <- -1
    } else {
        back <- 1
    }
    ls_t1 <- i - back
    while (landsat_bad[ls_t1] | modis_bad[ls_t1]) {
        ##  Reverse directions if we hit the beginning:
        if (ls_t1 == 1) {
            back <- -1
        }

        ls_t1 <- ls_t1 - back
        if (ls_t1 == i) ls_t1 <- i - back
    }
    m_t1 <- ls_t1

    ##  Determine next good Landsat scene in time series to
estimate from:
    ##      NOTE: The special case for estimating the first and
last scenes
    ##          of the series are handled:
    if (i == nlayers(landsat_sim)) {
        forward <- -1
    } else {
        forward <- 1
    }
    ls_t3 <- i + forward
    if (ls_t3 <= ls_t1) ls_t3 <- ls_t1 + forward
    while (landsat_bad[ls_t3] | modis_bad[ls_t3]) {
        ##  Reverse directions if we hit the end:
        if (ls_t3 == nlayers(landsat_sim)) {
            forward <- -1
        }

        ls_t3 <- ls_t3 + forward
        if (ls_t3 == ls_t1 | ls_t3 == i) ls_t3 <- ls_t1 + forward
    }
    m_t3 <- ls_t3

    modis_t1 <- modis[[m_t1]]
    modis_t2 <- modis[[i]]
    modis_t3 <- modis[[m_t3]]
    landsat_t1 <- landsat[[ls_t1]]
    landsat_t3 <- landsat[[ls_t3]]


    writeRaster(modis_t1, filename="./tmp/modis_t1.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
```

```
    writeRaster(modis_t2, filename="./tmp/modis_t2.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
    writeRaster(modis_t3, filename="./tmp/modis_t3.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
    writeRaster(landsat_t1, filename="./tmp/landsat_t1.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
    writeRaster(landsat_t3, filename="./tmp/landsat_t3.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)

    system2(command="./lib/StarFM/source/StarFM.exe",
args="./src/StarFM_config.txt", wait=TRUE)

    landsat_t2_sim <- raster("./tmp/landsat_t2_sim.envi")


    ##  Set any -32768 to NA values before writing:
    landsat_t2_sim[ landsat_t2_sim == -32768 ] <- NA
    landsat_sim[[i]] <- landsat_t2_sim[]

    ##  In our filled data set, set any missing Landsat pixels to
those
    ##      simulated via StarFM:
    landsat_filled[[i]] <- landsat[[i]]
    masked <- Which(is.na(landsat_filled[[i]]), cells=TRUE)
    landsat_filled[[i]][ masked ] <- landsat_sim[[i]][ masked ]


    png(file=paste0("./output/simulations/sim_", i, ".png"),
width=800, height=800)
        plot(landsat_sim[[i]], col=color_vec, zlim=z.lim,
frame.plot=TRUE, main=paste("Simulation at Step", i, "From
Steps", ls_t1, "and", ls_t3), xlab="Projected Map Coordinates",
legend=FALSE)
        plot(r.leg, col=color_vec, zlim=z.lim, frame.plot=FALSE,
add=TRUE)
        plot(study_area, border=rgb(0,0,0,0.5), add=T)
    dev.off()


    plot(landsat_sim[[i]], col=color_vec, zlim=z.lim,
frame.plot=TRUE, main=paste("Simulation at Step", i, "From
Steps", ls_t1, "and", ls_t3), xlab="Projected Map Coordinates",
legend=FALSE)
    plot(r.leg, col=color_vec, zlim=z.lim, frame.plot=FALSE,
add=TRUE)
    plot(study_area, border=rgb(0,0,0,0.5), add=T)

    pbStep(pb, step=NULL, label='Processed Layer')
}
pbClose(pb, timer=T)


writeRaster(landsat_sim,
filename="./output/Subset_Landsat_NDVI_16day_sim.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
```

```
writeRaster(landsat_filled,
filename="./output/Subset_Landsat_NDVI_16day_filled.envi",
bandorder='BSQ', datatype='INT2S', format="ENVI", overwrite=TRUE)
##  Or load them if already compiled:
landsat_sim <-
brick("./output/Subset_Landsat_NDVI_16day_sim.envi")
landsat_filled <-
brick("./output/Subset_Landsat_NDVI_16day_filled.envi")



#########################################################

##  Optional, create animation of series over time:
old_wd <- getwd()
setwd("./output/animation")
saveHTML({
    par(mfrow=c(1,4))
    for (i in 1:nlayers(landsat_sim)) {
        plot(modis[[i]], col=color_vec, zlim=z.lim,
frame.plot=TRUE, main="\nMODIS 13Q1 NDVI", xlab="Projected Map
Coordinates", legend=FALSE)
        plot(r.leg, col=color_vec, zlim=z.lim, frame.plot=FALSE,
add=TRUE)
        plot(study_area, border=rgb(0,0,0,0.5), add=T)

        plot(landsat[[i]], col=color_vec, zlim=z.lim,
frame.plot=TRUE, main=paste("Timestep", i, "\nLandsat TM/ETM+/OLI
NDVI"), xlab="Projected Map Coordinates", legend=FALSE)
        plot(r.leg, col=color_vec, zlim=z.lim, frame.plot=FALSE,
add=TRUE)
        plot(study_area, border=rgb(0,0,0,0.5), add=T)

        plot(landsat_sim[[i]], col=color_vec, zlim=z.lim,
frame.plot=TRUE, main="\nSimulated StarFM NDVI", xlab="Projected
Map Coordinates", legend=FALSE)
        plot(r.leg, col=color_vec, zlim=z.lim, frame.plot=FALSE,
add=TRUE)
        plot(study_area, border=rgb(0,0,0,0.5), add=T)

        plot(landsat_filled[[i]], col=color_vec, zlim=z.lim,
frame.plot=TRUE, main="\nFused Landsat/StarFM NDVI",
xlab="Projected Map Coordinates", legend=FALSE)
        plot(r.leg, col=color_vec, zlim=z.lim, frame.plot=FALSE,
add=TRUE)
        plot(study_area, border=rgb(0,0,0,0.5), add=T)

        ani.pause()
    }},
    img.name="ndvi",
    imgdir="./images",
    htmlfile = "./index.html",
  autobrowse = FALSE,
    title = "Fused Landsat and MODIS NDVI Timeseries, February
2000 - December 2015",
    description = "These data were extracted from Google Earth
Engine, and processed according to the script linked here.",
```

```
    ani.width=1024, ani.height=320
)
setwd(old_wd)


##  Plot overlapping series, based on a "clicked" point
##      in the plot window:
plot(landsat[[6]])
plot(study_area, border=rgb(0,0,0,0.5), add=T)
plot(poi, add=T)
zoom(landsat[[6]], new=F)
plot(study_area, add=T)
plot(poi, add=T)
i <- raster::click(landsat[[6]], n=1, id=T, xy=F)
pdf(file="./output/South_Site_3_Time_Series.pdf", width=10,
height=5)
plot(as.numeric(modis[i][]), col="grey", type="l",
ylim=c(0,15000), xlab="Timestep", ylab="NDVI (Scaled)",
main="MODIS, Landsat, StarFM, and Fused Time Series")
lines(as.numeric(landsat_sim[i][]), col="black", type="l")
lines(as.numeric(landsat_filled[i][]), col="blue", type="l")
lines(as.numeric(landsat[i][]), col="red", type="l")
legend("topleft",
    legend=c("MODIS","Landsat", "StarFM", "Fused"),
    col=c("grey", "red", "black", "blue"),
    lty=c(1,1,1,1),
    bg="white"
)
dev.off()
```

APPENDIX C: R PROGRAMMING CODE FOR BFAST

```
#install.packages("bfast", repos="http://R-Forge.R-project.org",
type = "source")
#update.packages(checkBuilt=TRUE)
require(bfast)

##  Install the development version of raster:
#install.packages("raster", repos="http://R-Forge.R-project.org")
#install.packages("raster", repos="http://R-Forge.R-project.org",
type="source")
require(raster)

require(TSA)
require(zoo)
#require(snow)
#install.packages("dplR")
#require(dplR)
#require(tcltk)     ##  For the progress bar
#require(xts)
require(rgdal)


#setwd("D:/Documents/Graduate
School/Research/Wallowa/Analysis/MODIS BFAST Analysis/")
#setwd("D:/Research/MODIS BFAST Analysis, Interactive,
Animation/")
setwd("C:/tmp/")

##  Load data:
#setwd("D:/Documents/Graduate
School/Research/Wallowa/Data/MODIS/R Work/output")
setwd("./output")

##  Load the Whitaker filtered output:
#vi_raster <- "EVI_YearlyLambda500_fullPeriod.tif"
#vi_raster <- "mod13q1_2000-2015_ndvi_pre_whittaker.gri"
##vi_raster <- "Subset_Landsat_NDVI_16day_filled.envi"
vi_raster <- "../data/South_Site_2_Subset_MODIS_NDVI_16day.tif"
##  Load the simple QC threshold, linear interpolated output:
##vi_raster_simple <- "mod13q1_2000-2015_ndvi_fixed.gri"
##vi_raster_simple <- "Subset_Landsat_NDVI_16day_filled.envi"
vi_raster_simple <-
"../data/South_Site_2_Subset_MODIS_NDVI_16day.tif"

##  Brick is very very much faster:
#raster_data <- stack(vi_raster)
raster_data <- brick(vi_raster)
raster_data_simple <- brick(vi_raster_simple)
```

```
##  Write out a layer from the raster_data converted to cell
indices for
##      comparing things by han#raster_indices <-
raster(vi_raster, band=1)
#raster_indices[] <- 1:length(raster_indices[])
#raster_indices <- writeRaster(raster_indices,
filename="mod13q1_raster_indices.gri", format="raster",
bandorder="BSQ", datatype="INT4U", overwrite=TRUE)
#raster_indices <- raster(vi_raster_simple)



##  Load date names:
#dates <- read.table("EVI_YearlyLambda500fullPeriod",
stringsAsFactors=FALSE)

##  Load Wallowa county boundary:
boundary <- readOGR("../data", "South_Site_2")
poi <- readOGR("../data", "South_Site_2_Point_Merge")


##  Set our layer names and write our break information stack
##      out to disk:
num_features <- 10
breaks <- 3
num_features_seasonal <- 10

layer_names <- character(length=(2 + num_features*breaks + 5 + 2
+ num_features_seasonal*breaks + 5))
layer_names[] <- ""
layer_names[1] <- "detected_breaks"
layer_names[2] <- "entropy"
for (i in 1:breaks) {
    layer_names[(i-1)*num_features+3] <-
paste("b",i,"_beg",sep="")
    layer_names[(i-1)*num_features+4] <-
paste("b",i,"_end",sep="")
    layer_names[(i-1)*num_features+5] <-
paste("b",i,"_len",sep="")
    layer_names[(i-1)*num_features+6] <-
paste("b",i,"_mean",sep="")
    layer_names[(i-1)*num_features+7] <-
paste("b",i,"_slope",sep="")
    layer_names[(i-1)*num_features+8] <-
paste("b",i,"_mean_diff",sep="")
    layer_names[(i-1)*num_features+9] <-
paste("b",i,"_break_diff",sep="")
    layer_names[(i-1)*num_features+10] <-
paste("b",i,"_25pct_ci",sep="")
    layer_names[(i-1)*num_features+11] <-
paste("b",i,"_break",sep="")
    layer_names[(i-1)*num_features+12] <-
paste("b",i,"_75pct_ci",sep="")
}
layer_names[(i)*num_features+3] <- paste("b",i+1,"_beg",sep="")
layer_names[(i)*num_features+4] <- paste("b",i+1,"_end",sep="")
```

```
521   layer_names[(i)*num_features+5] <- paste("b",i+1,"_len",sep="")
522   layer_names[(i)*num_features+6] <- paste("b",i+1,"_mean",sep="")
523   layer_names[(i)*num_features+7] <- paste("b",i+1,"_slope",sep="")
524
525   ##  Write out layer names for seasonal break detections:
526   j = (i)*num_features+7
527   layer_names[j + 1] <- "detected_breaks_seasonal"
528   layer_names[j + 2] <- "entropy_seasonal"
529   for (i in 1:breaks) {
530       layer_names[j+(i-1)*num_features_seasonal+3] <-
531   paste("sb",i,"_beg",sep="")
532       layer_names[j+(i-1)*num_features_seasonal+4] <-
533   paste("sb",i,"_end",sep="")
534       layer_names[j+(i-1)*num_features_seasonal+5] <-
535   paste("sb",i,"_len",sep="")
536       layer_names[j+(i-1)*num_features_seasonal+6] <-
537   paste("sb",i,"_range",sep="")
538       layer_names[j+(i-1)*num_features_seasonal+7] <-
539   paste("sb",i,"_entropy",sep="")
540       layer_names[j+(i-1)*num_features_seasonal+8] <-
541   paste("sb",i,"_range_diff",sep="")
542       layer_names[j+(i-1)*num_features_seasonal+9] <-
543   paste("sb",i,"_entropy_diff",sep="")
544       layer_names[j+(i-1)*num_features_seasonal+10] <-
545   paste("sb",i,"_25pct_ci",sep="")
546       layer_names[j+(i-1)*num_features_seasonal+11] <-
547   paste("sb",i,"_break",sep="")
548       layer_names[j+(i-1)*num_features_seasonal+12] <-
549   paste("sb",i,"_75pct_ci",sep="")
550   }
551   layer_names[j+(i)*num_features_seasonal+3] <-
552   paste("sb",i+1,"_beg",sep="")
553   layer_names[j+(i)*num_features_seasonal+4] <-
554   paste("sb",i+1,"_end",sep="")
555   layer_names[j+(i)*num_features_seasonal+5] <-
556   paste("sb",i+1,"_len",sep="")
557   layer_names[j+(i)*num_features_seasonal+6] <-
558   paste("sb",i+1,"_range",sep="")
559   layer_names[j+(i)*num_features_seasonal+7] <-
560   paste("sb",i+1,"_entropy",sep="")
561
562
563   ######
564   ##  Define a function to calculate the normalized spectral
565   entropy for a
566   ##      time series (as defined from the values extracted for
567   each
568   ##      pixel:
569
570   ##  As adapted from Zaccarelli (2013), variable names relate to
571   terms
572   ##      in the described Appendix and model:
573
574   ts_entropy <- function(x) {
575       ##      x is assumed to already be a timeseries object:
576       Ps <- spec(x, log="no", plot=FALSE)
577       Ps_spec <- Ps$spec
```

```
    Pk <- Ps_spec / sum(Ps_spec)
    Hsn_x <- -1 * sum(Pk * log(Pk)) / log(length(Pk))

    ##  The following would be the start to performing the
bootstrap for
    ##      confidence interval generation, but for now we'll
just return the
    ##      normalized spectral entropy (Hsn) from the series and
call it good.

    ### Now randomly resample the timeseries to create a
null/noise model:
    #x_null <- sample(x)
    #
    #Ps <- spec(x_null, log="no", plot=FALSE)
    #Ps_spec <- Ps$spec
    #Pk <- Ps_spec / sum(Ps_spec)
    #Hsn_null <- -1 * sum(Pk * log(Pk)) / log(length(Pk))

    return(Hsn_x)
}
######


######
##  Define a function to select and display BFAST calculation for
a particular pixel:

##  Define a function for calculating the significance vector
from a vector to create a series:

##  This could be sped up quite a bit by doing the COI creation
once beforehand and feeding it to the function as an argument:

plot_bfast_pixel <- function(view_raster=raster_data,
cell_id=NULL, extract_raster=raster_data, qc_raster=NULL,
classed=FALSE, rdist=NULL, season="harmonic", max.iter=1,
breaks=3, write_label=NULL) {
    #require(bfast)

    if (!is.null(write_label)) {
        sink(file=paste("./printed_", write_label, ".txt",
sep=""), append=TRUE, split=TRUE)
    }

    if (is.null(cell_id)) {

        if (extent(view_raster) != extent(extract_raster)) {
            quit("ERROR: Extents of view and extract rasters must
be equal!")
        }
        require(colorspace)

        print("Zoom to area:")
        flush.console()
```

```
        if (classed) {
            plot(view_raster[[1]], col=c(0,rainbow_hcl(7, c = 60,
l = 75)), breaks=0:8, zlim=c(0, 8))
          plot(boundary, add=T)
          plot(poi, add=T)
          zoom(view_raster[[1]], col=rainbow_hcl(7, c = 60, l =
75), breaks=1:8, zlim=c(1, 8))
          plot(boundary, add=T)
          plot(poi, add=T)

        } else {
            plot(view_raster[[1]])
          plot(boundary, add=T)
          plot(poi, add=T)
          zoom(view_raster[[1]])
            plot(boundary, add=T)
            plot(poi, add=T)

        }

        print("Choose pixel to calculate:")
        flush.console()


        pixel <- click(view_raster[[1]], n=1, id=TRUE, xy=TRUE,
cell=TRUE)

        cell_id <- pixel[[3]]
    }

    print(paste("Pixel", cell_id, "selected, calculating
BFAST...", sep=" "))
    flush.console()


    x <- extract(extract_raster, cell_id)


    ##  Check to see if we have a provided qc dataset and if so
replace
    ##      any bad values with NAs:
    ##  NOTE:  This is handled outside the bfast_pixel() function
in the full
    ##      raster processing...
    if (!is.null(qc_raster)) {
        qc <- extract(qc_raster, cell_id)
        x <- as.numeric(x)
        series <- ts(x, start=c(2000,2,18), deltat=16/365)

        ##  We want to replace any data in the original series
with an NA if the
        ##      converted QC data is 1 or NA:
        qc <- as.numeric(qc)
        qc[is.na(qc)] <- 1
        series_qc <- ts(qc, start=c(2000,2,18), deltat=16/365)

        ##  Do the replacement:
```

```
692             x[qc == 1] <- NA
693 
694             series_fix = ts(x, start=c(2000,2,18), deltat=16/365)
695 
696             ##  If we're set to write output to disk then create a
697 PDF of our output:
698             if (!is.null(write_label)) {
699                 pdf(file=paste("./plotted_qc_", write_label, ".pdf",
700 sep=""))
701                 plot(cbind(zoo(series), zoo(series_qc),
702 zoo(series_fix)), col=c("red", "blue", "green"))
703                 dev.off()
704             }
705             plot(cbind(zoo(series), zoo(series_qc), zoo(series_fix)),
706 col=c("red", "blue", "green"))
707         }
708 
709 
710         series <- ts(as.numeric(x), start=c(2000,2,18),
711 deltat=16/365, frequency=23)
712         series[series == -32768] <- NA
713 
714 
715         ##  Create an output data.frame() to hold info from the
716 analysis.  This
717         ##      should be one value for the number of trend
718 breakpoints detected,
719         ##      plus ten values for each break point plus five for
720 the segment
721         ##      after the last break point estimated where no
722 differences/CI can
723         ##      be estimated:
724         num_features <- 10
725         num_features_seasonal <- 10
726 
727         output <- numeric(length=(2 + num_features*breaks + 5 + 2 +
728 num_features_seasonal*breaks + 5))
729         output[] <- NA
730 
731         ##  Make sure we have more than one endpoint and then go:
732         if (sum(!is.na(series)) > 1) {
733             ##  Interpoloate any missing values:
734             series <- na.approx(series)
735 
736             ##  This is to set the minimum (in proportion of the
737 length of the total
738             ##      series) amount of time between detected features.
739 Verbesselt
740             ##      (in press) suggests at least an annual cycle
741 between detected
742             ##      breaks:
743             if (is.null(rdist)) {
744                 #rdist <- 10/length(series)
745 
746                 ##  One year minimum distance:
747                 #rdist <- 1 / (length(series) / 23)
748 
```

```
            ##  Two year minimum distance:
            rdist <- 2 / (length(series) / 23)
        }

        ##  Run bfast on our pixel time series:
        start_time = proc.time()[3]
        fit <- bfast(series, h=rdist, season=season,
max.iter=max.iter, breaks=breaks)

        ##  If we're set to write output to disk then create a
PDF of our output:
        if (!is.null(write_label)) {
            pdf(file=paste("./plotted_bfast_", write_label,
".pdf", sep=""))
            plot(fit)
            dev.off()
        }
        #plot(fit)
        bfast:::plot.bfast(fit)
        #bfast:::plot.bfast(fit, ANOVA=TRUE)
        #bfast:::plot.bfast(fit, type="seasonal")
        #bfast:::plot.bfast(fit, type="trend")
        #bfast:::plot.bfast(fit, type="all")

        print(paste("BFAST fit elapsed time:", proc.time()[3] -
start_time, "seconds"))
        flush.console()


        ##  Extract the first iteration values:
        iter <- 1
        out <- fit$output[[iter]]


        ##  Check trend component for breaks:
        detected_breaks <- out$Vt.bp

        ##  Check to see if any breakpoints were detected in the
trend component:
        if (detected_breaks[1] > 0) {
            output[1] <- length(detected_breaks)

            ##  To hand extract pieces of the trend based on
breakpoint locations:
            info <- out$ci.Vt
            times <- c(0, info$confint[,2], length(out$Tt))
        } else {
            output[1] <- 0
            times <- c(0, length(out$Tt))
        }


        ##  Store our entropy for the series:
        output[2] <- ts_entropy(series)
        print(paste("Norm. Spectral Entropy:",output[2]))

        for (i in 1:(length(times)-1)) {
```

```
806                 begTime <- times[i] + 1
807                 endTime <- times[i+1]
808 
809                 output[(i-1)*num_features + 3] <- begTime
810                 print(paste("Begin Time:",begTime))
811 
812                 output[(i-1)*num_features + 4] <- endTime
813                 print(paste("End Time:",endTime))
814 
815                 output[(i-1)*num_features + 5] <- endTime - begTime
816                 print(paste("Length:",endTime-begTime))
817 
818                 output[(i-1)*num_features + 6] <-
819   mean(out$Tt[begTime:endTime])
820                 print(paste("Means:",mean(out$Tt[begTime:endTime])))
821 
822                 output[(i-1)*num_features + 7] <- (out$Tt[endTime] -
823   out$Tt[begTime])/(endTime - begTime)
824                 print(paste("Slope:",(out$Tt[endTime] -
825   out$Tt[begTime])/(endTime - begTime)))
826 
827                 if (i != (length(times)-1)) {
828                     output[(i-1)*num_features + 8] <-
829   mean(out$Tt[((endTime + 1):times[i+2])]) -
830   mean(out$Tt[begTime:endTime])
831                     print(paste("Means Diff.:",mean(out$Tt[((endTime
832   + 1):times[i+2])]) - mean(out$Tt[begTime:endTime])))
833 
834                     output[(i-1)*num_features + 9] <- out$Tt[endTime
835   + 1] - out$Tt[endTime]
836                     print(paste("Diff. at Break:",out$Tt[endTime + 1]
837   - out$Tt[endTime]))
838 
839                     output[(i-1)*num_features + 10] <-
840   info$confint[i,1]
841                     print(paste("25% CI:", info$confint[i,1]))
842 
843                     output[(i-1)*num_features + 11] <-
844   info$confint[i,2]
845                     print(paste("Breakpoint:", info$confint[i,2]))
846 
847                     output[(i-1)*num_features + 12] <-
848   info$confint[i,3]
849                     print(paste("75% CI:", info$confint[i,3]))
850                 }
851                 flush.console()
852             }
853 
854 
855             ##  Write out layer names for seasonal break detections:
856             j = (breaks)*num_features+7
857 
858             ##  Check seasonal component for breaks:
859             detected_breaks <- out$Wt.bp
860 
861 
```

```
        ##  Check to see if any breakpoints were detected in the
seasonal trend
        ##      component:
        if (detected_breaks[1] > 0) {
            output[j + 1] <- length(detected_breaks)

            ##  To hand extract pieces of the trend based on
breakpoint locations:
            info <- out$ci.Wt
            times <- c(0, info$confint[,2], length(out$Tt))
        } else {
            output[j + 1] <- 0
            times <- c(0, length(out$Tt))
        }

        ##  Store our entropy for the smoothed, seasonal series:
        output[j + 2] <- ts_entropy(out$St)
        print(paste("Norm. Spectral Entropy of Seasonal Trend
Series:",output[j+2]))
        for (i in 1:(length(times)-1)) {
            begTime <- times[i] + 1
            endTime <- times[i+1]

            output[j+(i-1)*num_features_seasonal+3] <- begTime
            print(paste("Begin Time:",begTime))

            output[j+(i-1)*num_features_seasonal+4] <- endTime
            print(paste("End Time:",endTime))

            output[j+(i-1)*num_features_seasonal+5] <- endTime -
begTime
            print(paste("Length:",endTime-begTime))

            output[j+(i-1)*num_features_seasonal+6] <- (
max(out$St[begTime:endTime]) - min(out$St[begTime:endTime]))
            print(paste("Range (peak to trough):", output[j+(i-
1)*num_features_seasonal+6]))

            output[j+(i-1)*num_features_seasonal+7] <-
ts_entropy(out$St[begTime:endTime])
            print(paste("Seasonal Trend Piece
Entropy:",output[j+(i-1)*num_features_seasonal+7]))

            if (i != (length(times)-1)) {
                output[j+(i-1)*num_features_seasonal+8] <-
(max(out$St[(endTime + 1):times[i+2]]) - min(out$St[(endTime +
1):times[i+2]])) - (max(out$St[begTime:endTime]) -
min(out$St[begTime:endTime]))
                print(paste("Range Diff.:",output[j+(i-
1)*num_features_seasonal+8]))

                output[j+(i-1)*num_features_seasonal+9] <-
ts_entropy(out$St[(endTime + 1):times[i+2]]) - output[j+(i-
1)*num_features_seasonal+7]
                print(paste("Entropy Diff. at
Break:",output[j+(i-1)*num_features_seasonal+9]))
```

```
                output[j+(i-1)*num_features_seasonal+10] <-
info$confint[i,1]
                print(paste("25% CI:", info$confint[i,1]))

                output[j+(i-1)*num_features_seasonal+11] <-
info$confint[i,2]
                print(paste("Breakpoint:", info$confint[i,2]))

                output[j+(i-1)*num_features_seasonal+12] <-
info$confint[i,3]
                print(paste("75% CI:", info$confint[i,3]))
            }
            flush.console()
        }
    }

    if (!is.null(write_label)) {
        sink()
    }

    return(output)
}



##  Examples:

##  Basic usage:
#system.time(output <-
plot_bfast_pixel(view_raster=raster_data[[12]],
extract_raster=raster_data, qc_raster=qc_data))
##  Because data are already corrected we have no qc_data:
#system.time(output <-
plot_bfast_pixel(view_raster=raster_data[[12]],
extract_raster=raster_data))

##  Calculate for a specified pixel for both the smoothed and
less-smoothed
##      data:
#system.time(output <- plot_bfast_pixel(cell_id=70217,
extract_raster=raster_data_simple))
#system.time(output <- plot_bfast_pixel(cell_id=28822,
extract_raster=raster_data_simple))
#system.time(output <- plot_bfast_pixel(cell_id=70217,
extract_raster=raster_data))

##  Basic usage with selection from raster and output images and
text to a
##      file:
#output <- plot_bfast_pixel(view_raster=raster_data[[12]],
extract_raster=raster_data, qc_raster=qc_data,
write_label="land_1")
##  Because data are already corrected we have no qc_data:
#output <- plot_bfast_pixel(view_raster=raster_data[[12]],
extract_raster=raster_data, write_label="land_1")
write_label <- "South_Site_2_MODIS_BFAST"
```

```
output <- plot_bfast_pixel(view_raster=raster_data_simple[[6]],
extract_raster=raster_data_simple, write_label=write_label)
output_df <- data.frame(t(as.matrix(output)))
names(output_df) <- layer_names
write.csv(output_df, file=paste0(write_label, "_output.csv"))
```

APPENDIX D: MODIS BFAST PLOTS

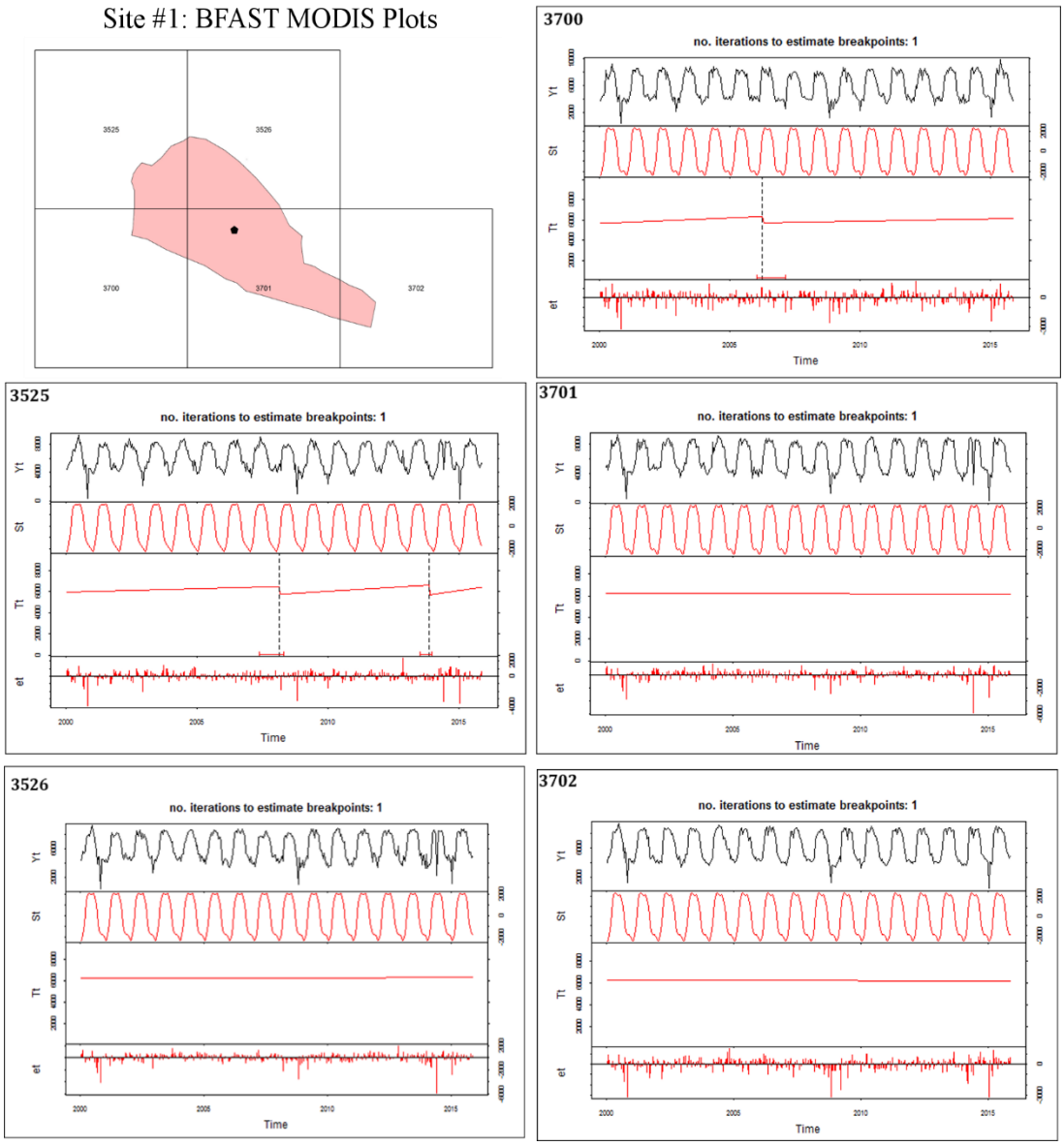

Figure D.1. BFAST plots for all MODIS pixels containing Site #1.

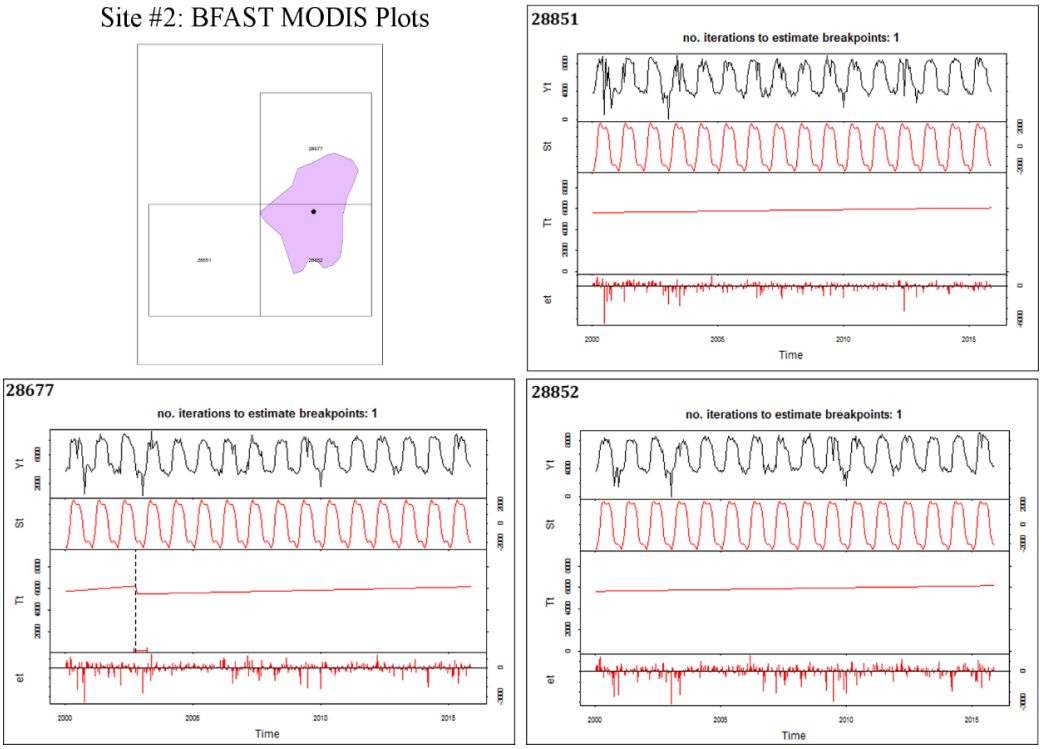

Figure D.2. BFAST plots for all MODIS pixels containing Site #2.

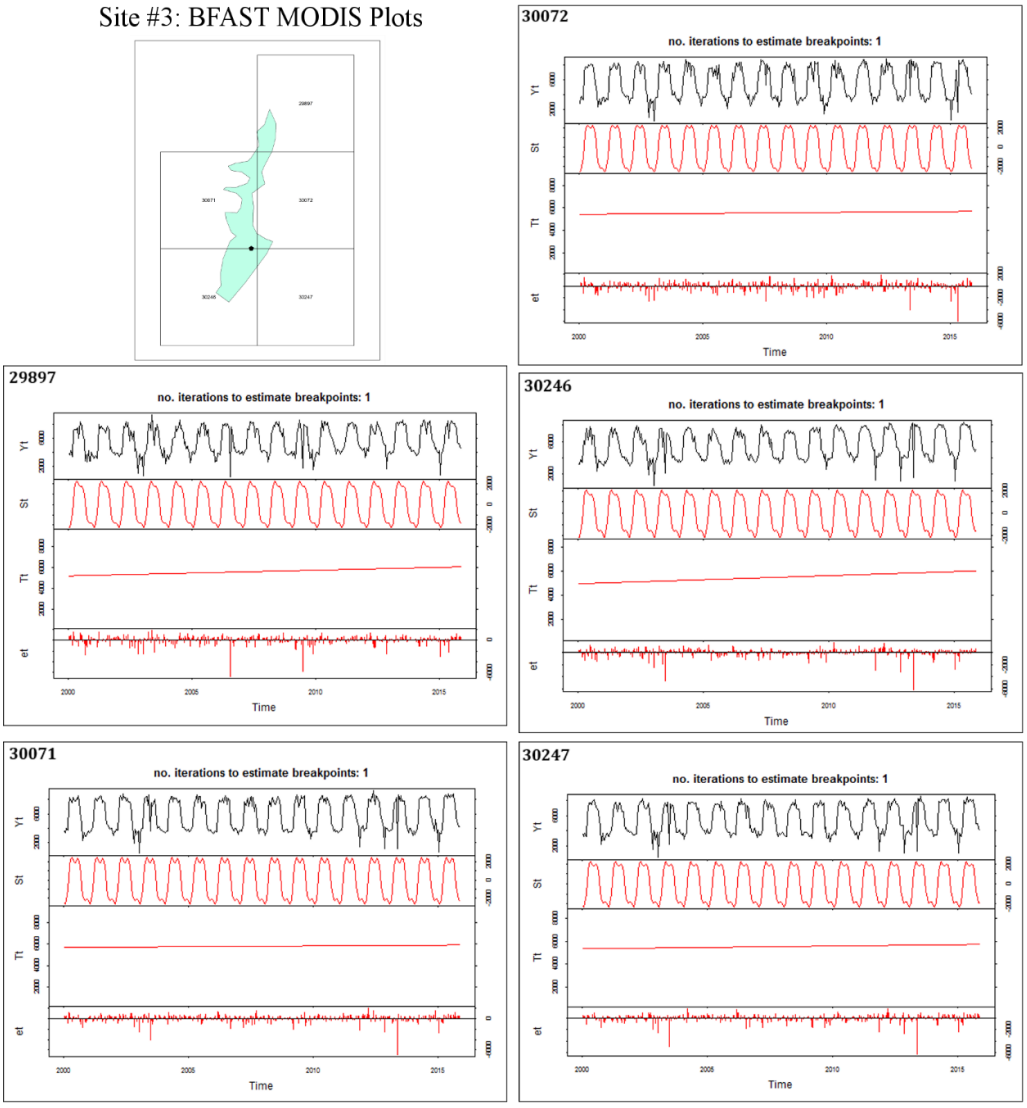

Figure D.3. BFAST plots for all MODIS pixels containing Site #3.

APPENDIX E: BFAST PLOTS USING FUSED IMAGERY FOR ALL POINTS

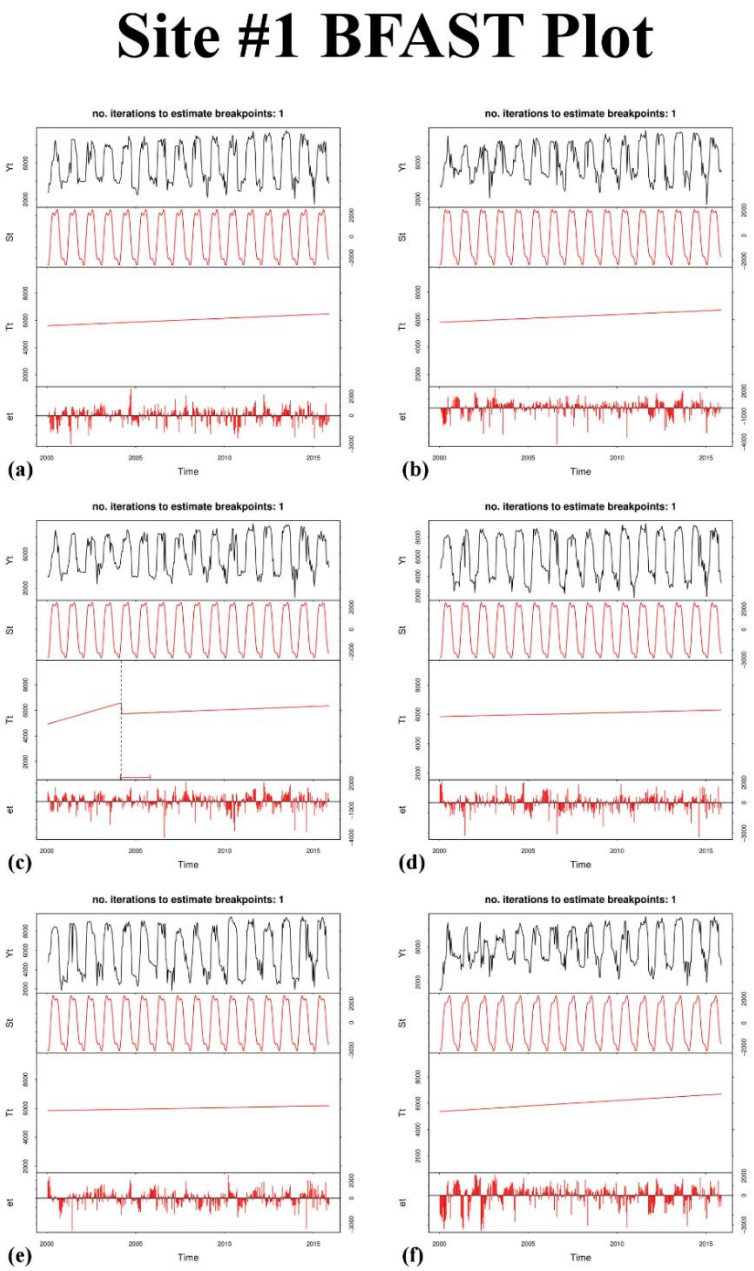

Figure E.1. Site #1 BFAST plots using fused Landsat-scale data for centroid (a) and random points 0-4 (b-f).

Site #2 BFAST Plot

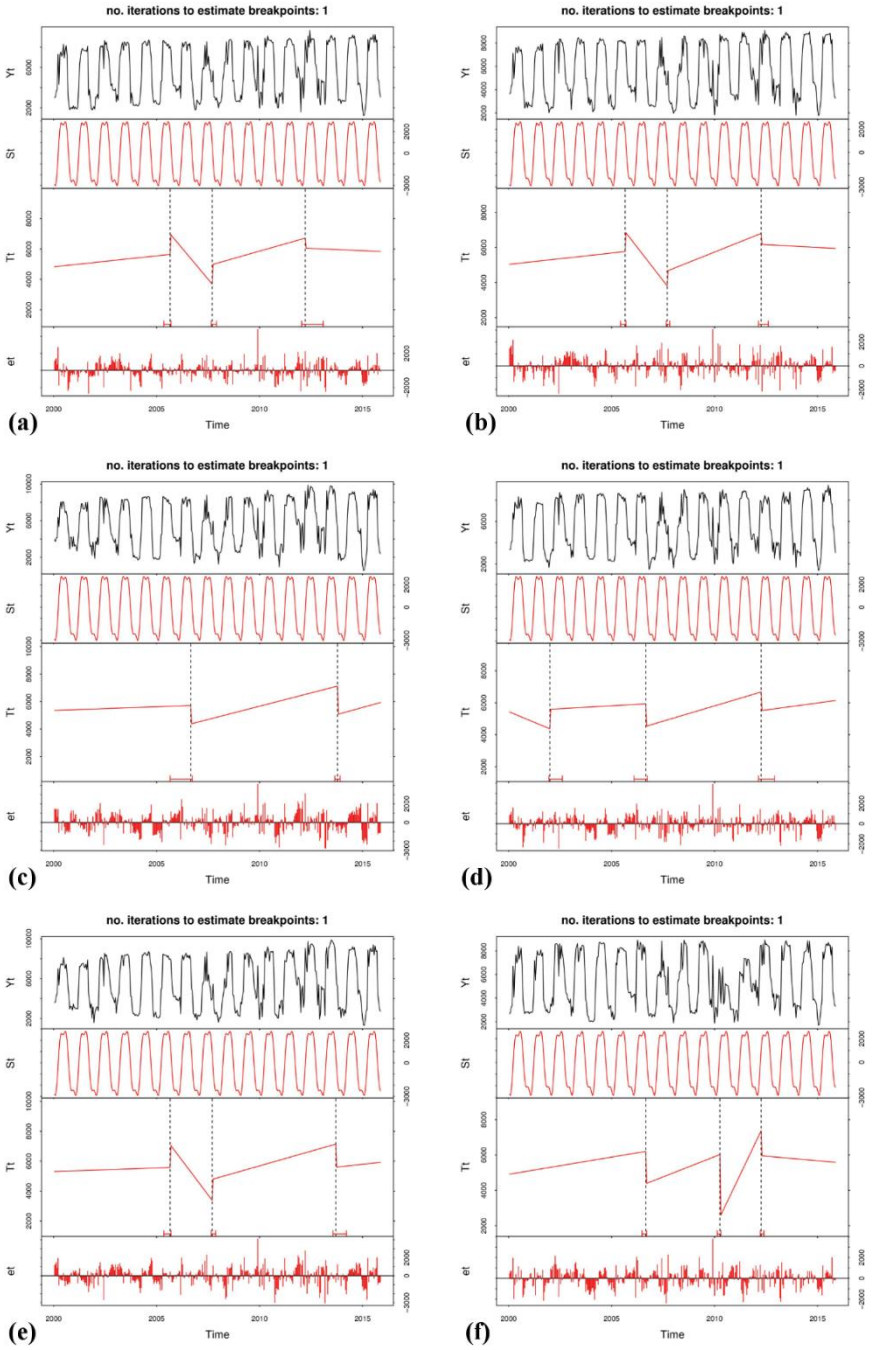

Figure E.2. Site #2 BFAST plots using fused Landsat-scale data for centroid (a) and random points 0-4 (b-f).

Site #3 BFAST Plot

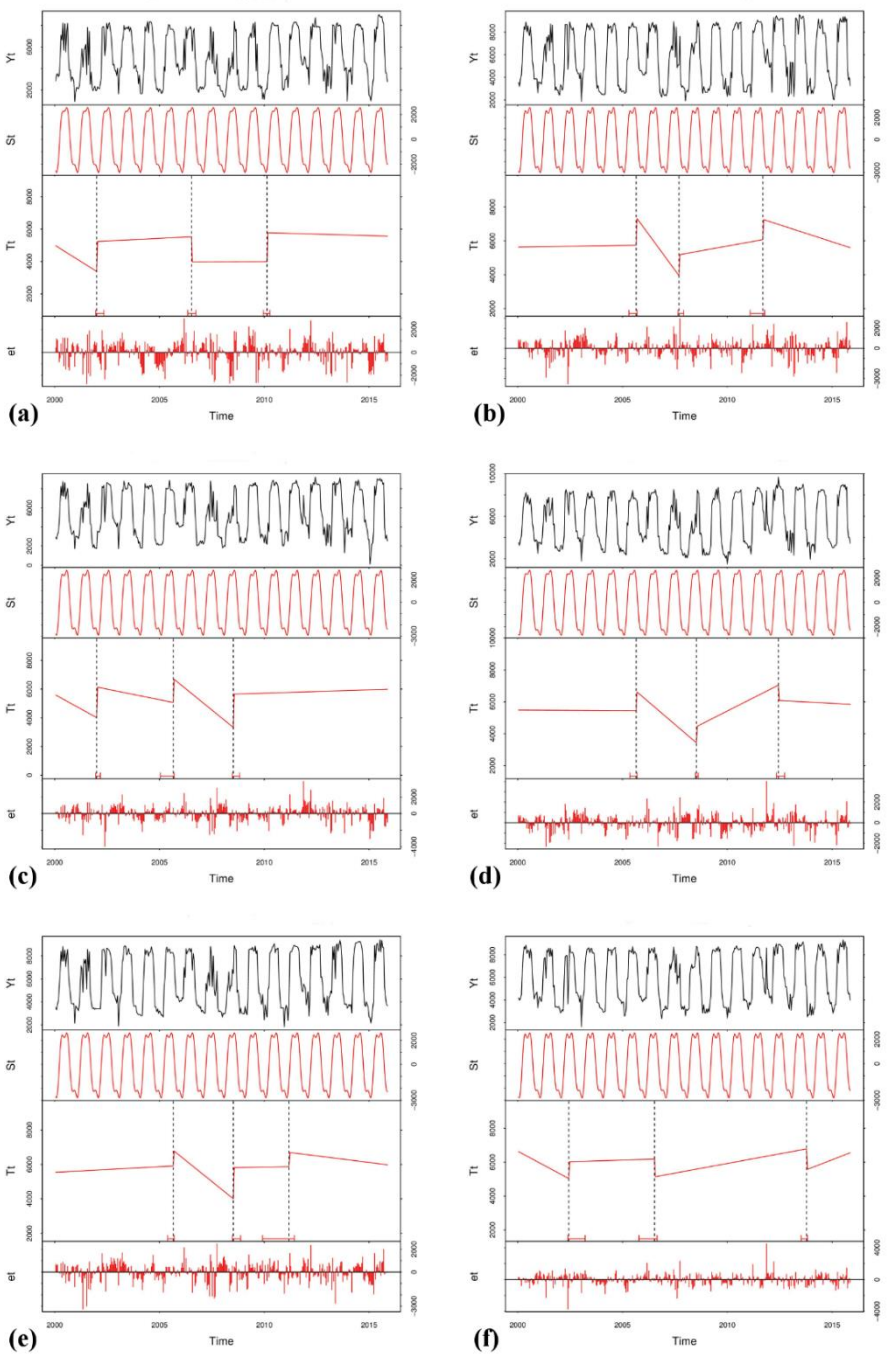

Figure E.3. Site #3 BFAST plots using fused Landsat-scale data for centroid (a) and random points 0-4 (b-f).

CURRICULUM VITA

NAME:	Faye Elizabeth Peters
ADDRESS:	Department of Geography and Geosciences 20 Lutz Hall University of Louisville Louisville, KY 40292
DOB:	Louisville, Kentucky – October 12, 1983
EDUCATION:	B. A., Earth and Planetary Sciences University of New Mexico, 2007
AWARDS:	Graduate Teaching Position University of Louisville, Geography and Geosciences 2014
	New Mexico Geological Society Fall Conference Scholarship University of New Mexico 2006
	Amigo/Amigo Scholarship University of New Mexico 2005
	General Thomas Campbell Award University of New Mexico 2005
	National Student Exchange University of Louisville 2003
PROFESSIONAL SOCIETIES:	Association of American Geographers 2015

TEACHING EXPERIENCE:

University of Louisville
Geoscience 200: The Global Environment (online course)
2015-2016

Graduate Teaching Assistant
University of Louisville, Geography and Geosciences
2014-2015