Introduction

The terrestrial ecosystem and its vegetation exerts great impact on the planet due to its role in biogeochemical cycles (Hansen & Loveland, 2012). This impacts society in terms of ecosystem services in its many forms locally and globally (ibid.). From this there is of great interest to study and monitor vegetation (ibid.). The costs linked to rigorous ecosystem monitoring can imply that the quality of data to be various across the world which can prove cumbersome to handle when conducting global studies (Friedl et al, 2002). Remote sensing technologies have resulted in better uniform data to be used for mapping of global vegetation instead of relying on potential vegetation maps produced from climate variables such as temperature from atlases for instance (Friedl et al, 2002). Remote sensing also makes it possible to monitor the dynamic aspects of ecosystems as changes can be observed as in-situ measurements can be irrelevant if a change has occurred after the measurement (ibid.).

A popular product from remotely sensed data for monitoring vegetation is vegetation indices (Huete et al, 2002). Vegetation indices are transformation of spectral data, compromising of one or more spectral bands, to differentiate and highlight the amount and spatial extent of vegetation (ibid.). Several vegetation indices exist where each is calculated differently and has its shortcomings where the goal is often to estimate biophysical variables such as biomass (ibid.). A common vegetation index is the normalized vegetation index (NDVI) which is generally complemented to be chlorophyll sensitive but suffers from saturation at high vegetation biomass (Jin, H. & Eklundh, L., 2014). From this there is of high relevance to expand the catalogue of vegetation indices to rectify the shortcomings of the different vegetation indices or to complement them. A new promising vegetation index is the plant phenology index (PPI) which is a vegetation index that is based upon radiative transfer equations and is highly correlated to the biophysical variable leaf area index (LAI) (ibid.). Vegetation indices can be a provided product as in the case of NDVI with the Moderate Resolution Imaging Spectroradiometer (MODIS) sensor but depending on index must be calculated by the user (Huete et al, 2002). In the popular geographic information system program ArcGIS there are implemented tools to calculate NDVI rather easy but there are none for PPI. In this report we showcase a new user-friendly tool to calculate PPI for various end user purposes built using the python programming language in conjunction with the ArcGIS tool builder.

Methodology

Calculation of PPI

As PPI is an integral part of the tool one must know how it was implemented. The equation of PPI is relatively straight forward but can require certain assumptions of specific variables. The PPI equation is as follows:

$$PPI = -K * \ln\left(\frac{M - DVI}{M - DVI_S}\right) \quad (eq. 1)$$

Here *K* is the gain factor, *DVI* is the differential vegetation index computed as the difference between the near infrared band and the red band, *DVI*$_S$ is the *DVI* of the soil and *M* is the site-specific canopy maximum DVI (Jin & Eklundh, 2014). *DVI*$_S$ is set to 0.09 in the tool which is based upon the value used in Jin & Eklundh (2014) derived from reflectance of 41 soil samples. *M* can be estimated in several ways where the most prominent method is a value based upon several years of measurements for the site to be studied (ibid.) However, here *M* is set at a default value of 0.5 in this tool as according to H. Jin (Personal communication, 16 October 2017) it is a sufficient standard value if no other estimation is viable. *K* is calculated as follows:

$$K = \frac{1}{4Q_E} \times \frac{1 + M}{1 - M} \times \frac{\varphi}{-Ln(1 - \varphi)} \quad (eq. 2)$$

Here *Q*$_E$ is defined as the light extinction efficiency and φ is the leaf filling factor. However, since the last term containing φ according to Jin & Eklundh (2014) is usually 1 that term was omitted in the tool. Hence *K* in the tool is defined as:

$$K = \frac{1}{4Q_E} \times \frac{1 + M}{1 - M} \quad (eq. 3)$$

Q_E is calculated as follows:

$$Q_E = d_C + (1 - d_C) \times \frac{G}{\cos(\theta)} \quad (eq.4)$$

Here *G* is a function of leaf angular distribution and is set at 0.5. According to Jin. & Eklundh (2014) leaf structure defines *G* and a value of 0.5 is valid for both flat or needle leaf thus making that value a sufficient generalization. Furthermore, θ is defined as the solar zenith angle and d_c is defined as the instantaneous diffuse fraction of solar radiation at solar zenith angle θ. d_c is an empirical equation calculated as follows:

$$d_C = 0.0336 + \frac{0.0477}{\cos(\theta)} \quad (eq.5)$$

Script/tool structure

The ArcGIS tool is made from two specific components. One being the script, coded in python, from which the processing of the data is done and the other being ArcGIS tool builder with its own configurations which makes up the graphical user interface (GUI) in this case. As of this the tool to calculate PPI is packaged as an .tbx file to be used as an importable tool in ArcGIS with an embedded script. The data used to test and develop the script was Sentinel 2-B level-1C and Satellite Pour l'Observation de la Terre 5 (SPOT5) data over Skåne in Sweden.

In the tool window the user will be faced with various options of inputs ranging from separate NIR and Red band raster files or a single raster dataset containing multiple bands as in the case of SPOT. The tool is scripted such that depending on the choice of sensor specified by the user different fields will be activated to reduce potential input errors. For instance, if Sentinel is chosen as a sensor then two fields will be activated prompting input of a NIR and Red band respectively and if SPOT is chosen then a single field will prompt the user to input a single raster dataset. The user will also be prompted to chose a method to input a solar zenith angle where options to recieve it from metadata is available (only for Sentinel and SPOT), manual input or calculation of it within the tool interface if a user has the necessary values needed to calculate solar zenith angle. Furthermore, the *M* value which can be observed in equation 1 can be changed if needed and also the scaling factor. The scaling factor refers to the case that reflectance data can be delivered as an integer format from data providers with various ranges which has to be converted to reflectance values between 0 and 1. Lastly the user will chose if a seperate NDVI or enhanced vegetation index 2 (EVI2) should be created derived from the same data that PPI is calculated from. According to H. Jin (Personal communication, 16 October 2017) PPI is a relatively new vegetation index thus comparing PPI values to other vegetation indicies, in this case NDVI and EVI2, is highly recommended. The choice of NDVI and EVI2 stems from personal recommendation from H. Jin (Personal communication, 16 October 2017) due to their widespread use within the remote sensing community. If the user chose to create NDVI then two outputs will be created consisting of a NDVI raster and a NDVI differance raster computed as the difference of PPI and NDVI in order to look how certain pixels varies with vegeation index. This also applies to EVI2.

These option that the user will be faced with serve as triggers prompting the execution of certain if-statements and other processing steps in the script. This means that the flow and order of the script will change depending on what kind of input parameters are specified within the tool GUI. For instance, if metadata is chosen as solar zenith angle aquisition method then the implemented solar zenith calculation will not be run but a segment containing metadata read and extraction of solar zenith angle will. However, if a general scheme is to be provided for the flow of the script it would go as follows.

1. Initilizing of input parameters from the tool GUI within ArcGIS such as input raster path names specified by the user or solar zenith angle method for instance.
2. Aquisition of solar zenith angle from either metadata, manual input or calculation depending on step 1.
3. NIR and Red waveband aquisition from either direct path names to specific seperate files or by transforming a single raster dataset depending on step 1.
4. Calculation of DVI and PPI.

5. Calculation of NDVI and EVI2 if specified.

All raster outputs that are saved are automatically saved as a .tif file. This in order to be able to be processed by other GIS software not linked with ArcGIS as the default save extention is in a ArcGIS grid format. Hence, if a output name is named *PPIoutput* it will be saved as *PPIoutput.tif* by the tool. At the side a more simpler tool is available that only has manual input of Red and NIR band with manual input of solar zenith or calculation of it. This is due to some users might not want the sensor options etc.

Results

In this section Sentinel-2B data will be shown to present tool functioning. The data used for this purpose was Sentinel-2B data with acquired sensing time of 2017-10-12 on southern Scania in Sweden. The raster has a 40.1% cloud coverage with a processing level of 1C that equals to reflectance values in integer format that need to be scaled and thus divided by the delivered quantification value (Drusch et al, 2012). The NIR and Red band share the same pixel resolution of 10 meters (ibid.).

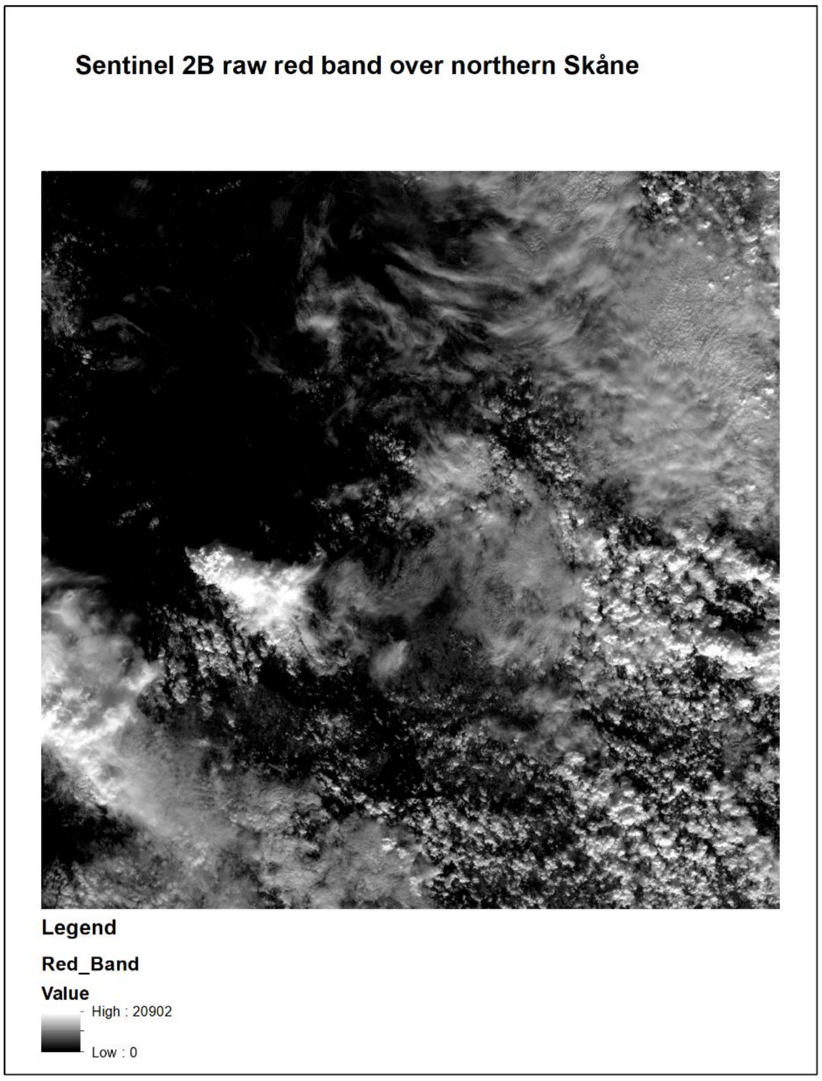

Figure 1: Showing the raw Red band over northern Skåne in Sweden. Sentinel-2B level-1C product gathered 2017-10-12.

Figure 1 shows the raw data from the red band over Scania. Here it can be seen that the integer values range from 0 to 20902 which when rescaled equals to reflectance values well above 1. The higher values in the range are dominated by various clouds as in the case of the cloud in the left centre of the raster. The lowest values are accommodated to the ocean as can be seen in the left side of the raster.

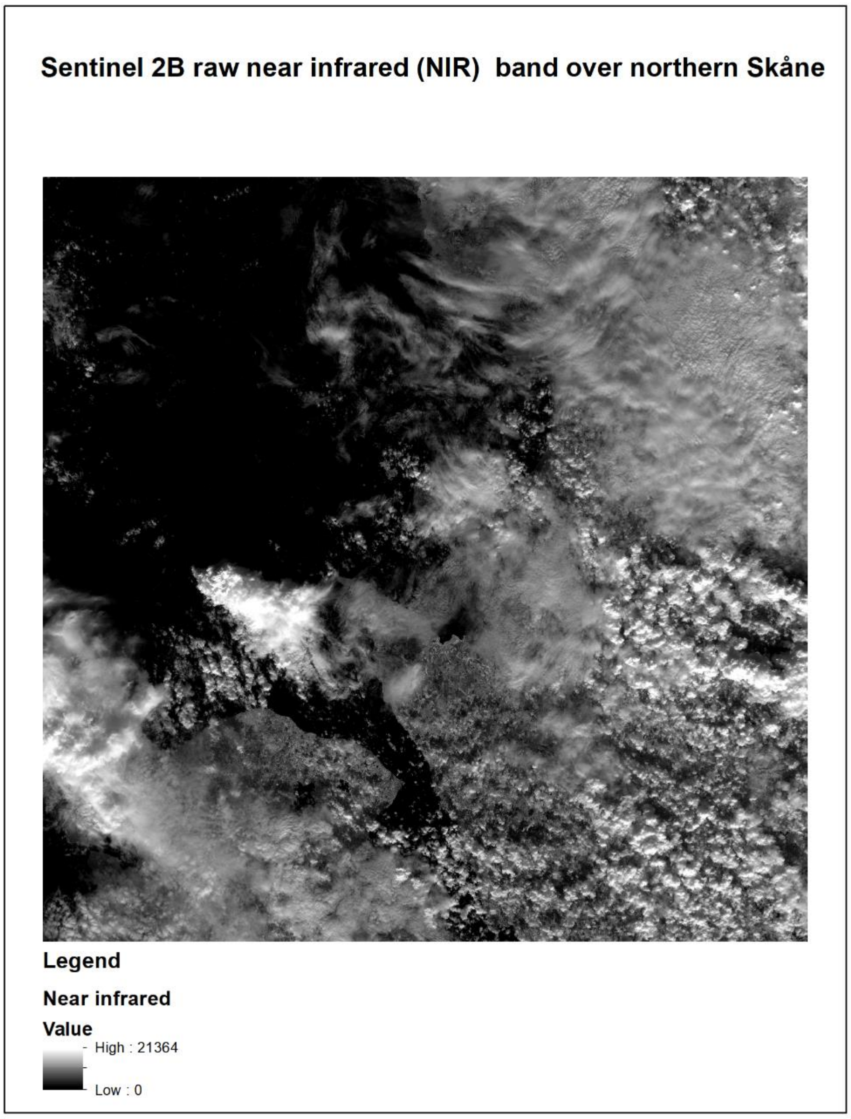
Sentinel 2B raw near infrared (NIR) band over northern Skåne
Legend
Near infrared
Value
High : 21364
Low : 0

Figure 2: Showing the raw NIR band over Skåne in Sweden. Sentinel-2B level-1C product gathered 2017-10-12.

In figure 2 the NIR band is shown with similar values to the red band in terms of reflectance spanning from 0 to 21364. The higher values of the range are dominated by various clouds especially in the left centre of the raster. The lowest values are situated in the left side of the raster that is ocean. Compared to figure 1 here land is clearly more visible.

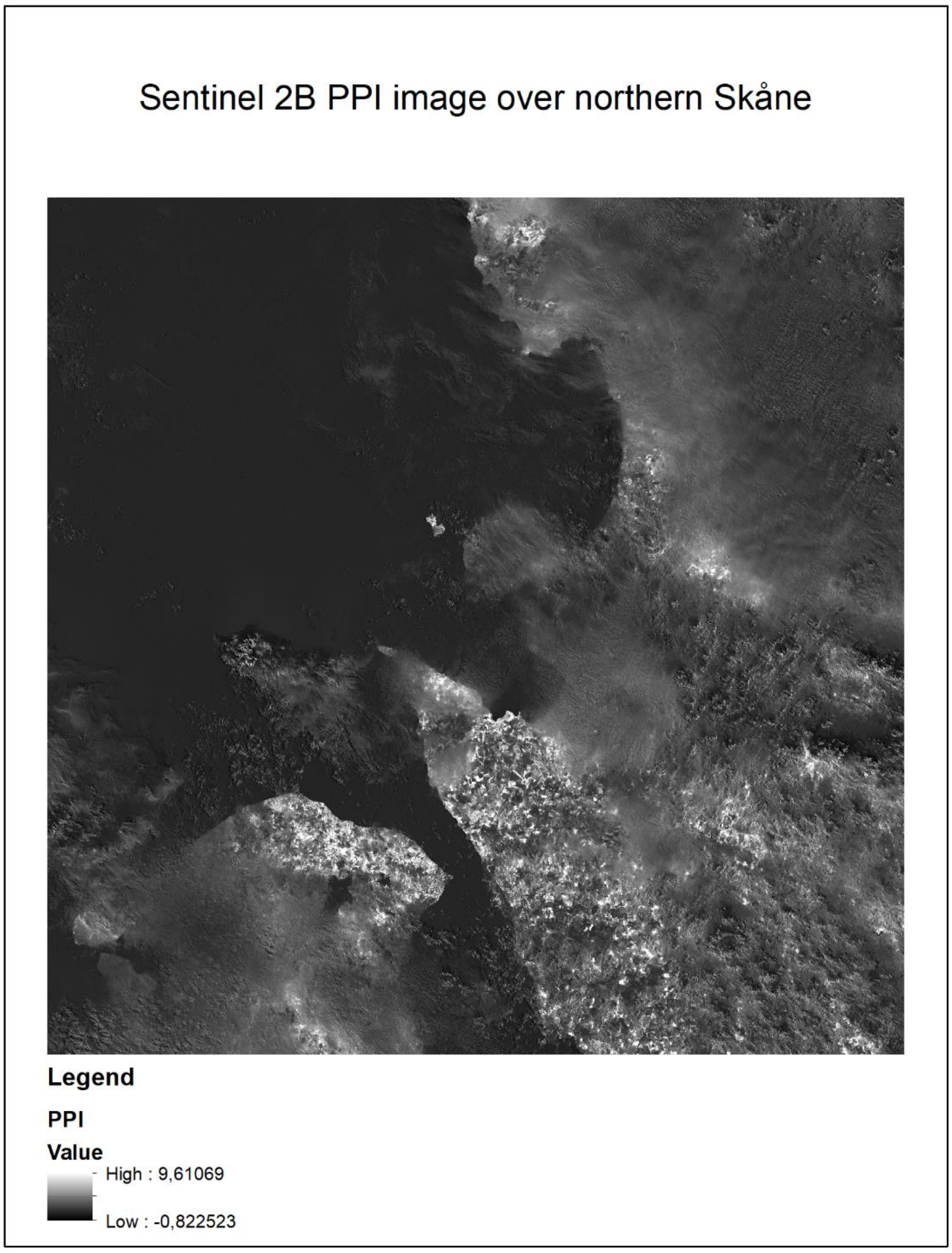

Figure 3: Showing calculated PPI using the tool created in this study using Sentinel-2B level-1C gathered 2017-10-12.

In figure 3 we the output PPI of the tool as a product of the raster images shown in figure 1 and 2. Here PPI values range from -0.82 to 9.61. Here the higher values are situated on land where the lower values are accommodated to clouds and water in general.

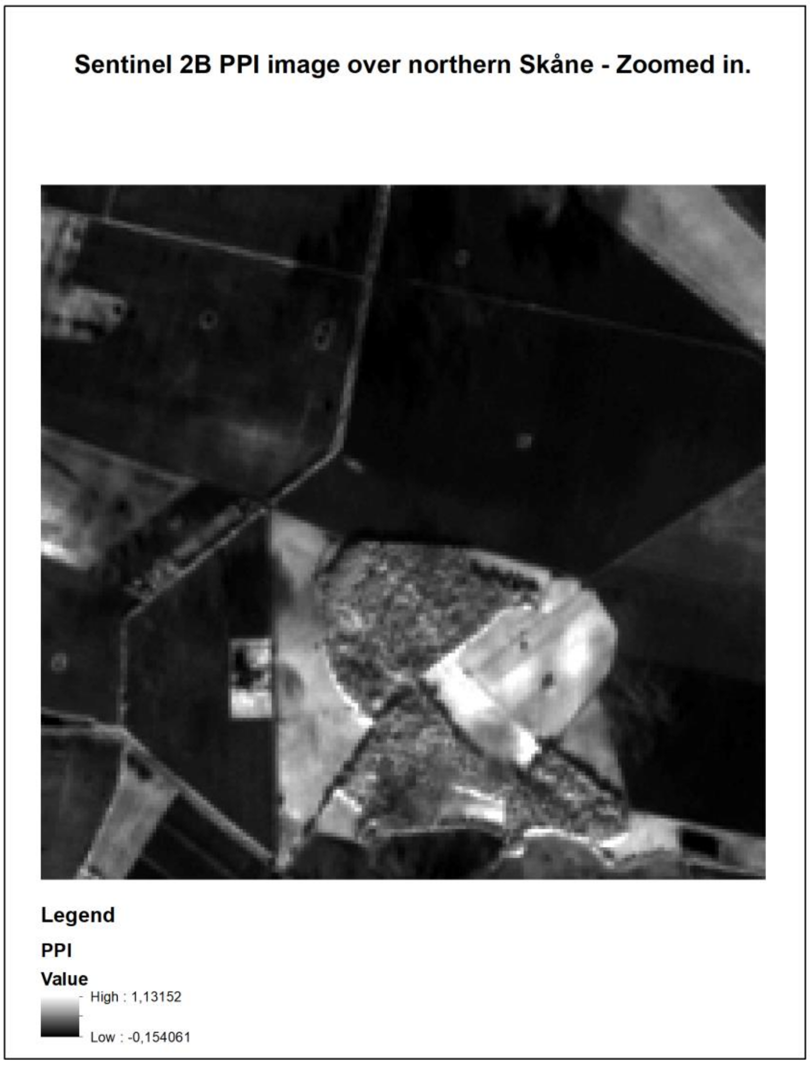

Figure 4: Showing calculated PPI zoomed in on an agricultural field using the tool created in this study using Sentinel-2B level-1C gathered 2017-10-12.

In figure 4 we see a zoomed in picture of the raster from figure 3. PPI Values range from -0.15 to 1.13. Here various agricultural fields are visible as bright in the centre of the raster with high values contra to the lower values in the periphery. The brightest field in the centre of the raster has PPI values around 0.65 while the forest at its left has PPI values around 0.15.

Discussion

As can be seen by the figures 3 and 4 the tool created in this study does successfully compute PPI values. However, there are problems that need to be mentioned and discussed. One point is the notion of the, in this case Sentinel data, high reflectance values. In figure 4 the high reflectance values above 1 only amount to 2 specific pixels within the image. This in turn could result from various atmospheric effects such as diffuse radiance (Chuvieco, 2016, p.235). This problem implies that there should be an implementation in the script to omit reflectance values, after they have been scaled accordingly, over 1. The importance of this lies in the fact that if reflectance values over 1 are not omitted then calculated PPI values cannot be trusted for the pixels in question. From this one could argue to simply remove all PPI values above 1 after the tool run but that could still mean that PPI values under 1 could be a product of reflectance values above 1. To prevent this potential misunderstanding omission of high reflectance values within the tool could be a possible solution but also some kind of input error system prompting the user to remove these outliers before running the tool.

However, this raises a question that has been highly impactful in the development of the tool which corresponds to exactly how much the tool should do for the user. For example, there is no function within the tool that does any kind of geometric correction of the raster input but on the other hand the user can scale the data accordingly or calculate the solar zenith. This makes the tool deviate from being a pure PPI calculator but to a more general tool. If it were more of a pure PPI calculator tool, then the user would have to scale the data accordingly manually and unpack raster datasets to acquire a separate NIR and Red raster. This can invoke certain questions why some functions are implemented and some not and why the tool “holds the users hand” in some sense. Having the tool being subject to testing by users, experienced or non-experienced, to receive tool input could greatly improve the tool functioning while also potentially remove unnecessary functions – making the tool more efficient. This has not been done and thusly the tool is an object purely created what the authors deemed to be important which does not have to be the case.

As the tool is created to handle manual but also sensor specific input questions arise when it comes to how exactly compatible the tool is to other data from the same sensor. For example, the Sentinel-2B data that was used to develop the function to find metadata solar zenith angle was specifically the level-1C product. Sentinel does however provide 5 different levels of products ranging from the very raw non-radiometrically/geometrically level-0 to the heavily computed bottom of the atmosphere reflectance (BOA) found in level-2A (Drusch et al, 2012). This could potentially mean that the solar zenith angle acquisition from metadata might not be valid for other kinds of product levels as the metadata file might look different. One could argue therefore that the data used to develop the script that runs the tool is quite narrow. A broader set of data from various sensors with different product levels should have been used to satisfy tool data compatibility. This however is not a problem if the user specifies manual input when prompted which sensor is used as the NIR and Red raster could originate from UAV and the solar zenith angle from this could be computed manually within the tool.

Lastly there is a lack of validation of PPI values within the tool. As Jin & Eklundh (2014) states, PPI needs good reflectance data as in sun sensor geometry corrected which corresponds to data with bi-directional reflectance difference function (BRDF) adjusted reflectance. However, if the user inputs data that is not corrected in this fashion PPI values can be misleading hence a kind of validation of PPI values against common values could be presented to the user. An implementation of a check if the calculated PPI values are within norms of PPI values could provide this. The included function to see the difference in PPI in relation to NDVI or EVI2 can serve this purpose of validation but it could be that it is not sufficient for the user with little to or no experience in PPI or vegetation indices in general.

As a conclusion the created tool with its script does work with various input that keeps a degree of freedom for the user to change input parameters. However, there are problems with the script as in (1) no management and information about outliers, (2) no user testing of the software, (3) no guarantee as to which data products are supported as in the case of Sentinel and SPOT other than those used in creation of the tool and (4) lastly that there is no kind of concrete easy to understand validation of calculated PPI values.

References

Chuvieco, E. (2016). *Fundamentals of Satellite Remote Sensing: An Environmental Approach*. Boca Raton, Florida: CRC Press.

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., … Bargellini, P. (2012). Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sensing of Environment*, *120*, 25-36.

Friedl, M.A., McIver, D.K., Hodges, J.C.F., Zhang, X.Y., Muchoney, D., Strahler, A.H., … Schaaf, C. (2002). Global land cover mapping from MODIS: algorithms and early results. *Remote Sensing of Environment*, *83*, 287-302.

Hansen, M.C., & Loveland, T.R. (2012). A review of large area monitoring of land cover change using Landsat data. *Remote Sensing of Environment*, *122*, 66-74.

Jin, H., & Eklundh, L. (2014). A Physically based vegetation index for improved monitoring of plant phenology. *Remote Sensing of Environment*, *152*, 512-525.

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

Appendix

```
 # -*- coding: utf-8 -*-
"""
Created on Sun Oct 15 14:52:22 2017

@author: Karl Adler, Simon Nåfält and Andreas Kroné.
"""


import arcpy
import math
arcpy.env.overwriteOutput = True


#The user specifies which sensor is used or if it is manual, from this different if statements will
#be executed depending on the choice.
sensor = arcpy.GetParameterAsText(0)


# If Sentinel or manual input is chosen then NIR and Red are defined by these directory inputs.
if str(sensor) == "Sentinel" or str(sensor) == "Other (manual input)":
   Red = arcpy.Raster(arcpy.GetParameterAsText(1)) #Input NIR raster, in tool represents the
pathway to the file
   NIR = arcpy.Raster(arcpy.GetParameterAsText(2)) #Input Red raster, in tool represents the
pathway to the file


if str(sensor) == "SPOT":
   Clumped_bands = arcpy.GetParameterAsText(3) #If there is only on main file then this will be used
as input


m = float(arcpy.GetParameterAsText(4)) #The m value as a string input
scale_fac = float(arcpy.GetParameterAsText(5)) #Scale factor is set at default 1.
zenith_Method = arcpy.GetParameterAsText(6) #A string which specify which method to use for
zenith method.
Zenith = arcpy.GetParameterAsText(7)
metadata = arcpy.GetParameterAsText(8)
```

```
# If the method of getting solar zenith is set to calculation then run these expressions.
if str(zenith_Method) == "Calculation":
    lat = float(arcpy.GetParameterAsText(9)) #Latitude input
    jul_date = float(arcpy.GetParameterAsText(10)) #Julian date input
    time_h = float(arcpy.GetParameterAsText(11)) #time of day input, 0-24, 12 as default for MODIS

if str(zenith_Method) == "Calculation":
    sun_dec = -23.45*math.cos((jul_date+10)*360/365*math.pi/180) #Sun declination
    h_ang = abs((time_h-12)*15)

    Zenith = math.acos(math.sin(math.pi/180*sun_dec) * math.sin(math.pi/180*lat) -
math.cos(math.pi/180*sun_dec) * math.cos(math.pi/180*lat) * math.cos(math.pi/180*h_ang)) *
180/math.pi

# Variables for checking if the users wants EVI2 and NDVI but also the output directory.
EVI2_checked = arcpy.GetParameterAsText(12) #In the tool represents a checkbox of either false or
true value
NDVI_checked = arcpy.GetParameterAsText(13)
output_test = arcpy.GetParameterAsText(14) #The output PPI pathway as a string

# Functions to derive the solar senith from SPOT or Sentinel MetaData if the user specified.
def firstZenith(txt, substring):
    with open(txt,'r') as f:
        strings = f.readlines()

    index = -1
     # following line finds first occurrence of the specified substring (in our case the target for
zenith/elevation).
    index = next(i for i, string in enumerate(strings) if substring in string) # if error says "StopIteration",
it means no matching substring was found
    if index == -1:
        index = 'error'                # if acceptable index is found
    return index
```

```
def solZen(dataSource,metaData):
    with open(metaData,'r') as f:
        mData = f.readlines()
        if dataSource == "SPOT":
            temp = mData[firstZenith(metaData,'SUN_ELEVATION')] # Same as above but for SPOT data
            solarZenith = 90.0-float(temp[15:-17]) # Solar zenith from sun elevation

        elif dataSource == "Sentinel":
            temp = mData[firstZenith(metaData,'<ZENITH_ANGLE')] # Same for Sentinel data
            solarZenith = float(temp[33:-16]) # Extracts the value from the string

    return solarZenith

# Calling on solZen function in order to obtain solar zenith angle from metadata file if solar zenith method is set to it
if str(zenith_Method) == "MetaData":
    Zenith = solZen(sensor,metadata)

# Import of a clumped file of bands as in the case of raw SPOT data to aquire NIR and Red data.
if str(sensor) == "SPOT":
    inp = Clumped_bands

    #Selecting Red and NIR bands from input SPOT-image and creating seperate temporary layer files
    Red = arcpy.MakeRasterLayer_management(inp,"Red","","","2")
    NIR = arcpy.MakeRasterLayer_management(inp,"NIR","","","3")

    #Saving temporary files to persistent layer files
    Red_lyr = arcpy.SaveToLayerFile_management(Red,output_test+"Red")
    NIR_lyr = arcpy.SaveToLayerFile_management(NIR,output_test+"NIR")

    #Converting layer files to raster files
    arcpy.ExtractSubDataset_management(Red_lyr,output_test+"Red")
    arcpy.ExtractSubDataset_management(NIR_lyr,output_test+"NIR")


    NIR = arcpy.Raster(output_test+"Red")
    Red = arcpy.Raster(output_test+"NIR")


# Scaling of the bands if specified as the calculations needs reflectance values from 0 to 1.
Red = Red*scale_fac
NIR = NIR*scale_fac


# Calculation of the differential vegetation index, to be used in PPI
NIR = arcpy.sa.Float(NIR)
Red = arcpy.sa.Float(Red)
DVI = NIR - Red


# Calculation of PPI.
dc = (0.0336 + 0.0477)/math.cos(Zenith) #Calculation of the instantaneuous diffuse fraction of solar
radiation at sun zenith angle
Qe = dc + (1-dc) * (0.5/math.cos(Zenith)) #Calculation of the canopy light extinction efficiency
K = (1/(4*Qe))*((1+m)/(1-m)) #Calculation of the gain factor used for PPI


PPI_1 = (m-DVI)/(m-0.09)
PPI = -K * arcpy.sa.Ln(PPI_1) #Calculation of PPI
PPI.save(output_test+".tif")


if str(EVI2_checked) == 'true': #If the EVI2 box is checked then this if statement will begin.
    EVI2 = 2.5*((NIR-Red)/(NIR+2.4*Red+1))
    EVI2.save(output_test+"EVI2.tif")
    EVI2diff = PPI - EVI2
    EVI2diff.save(output_test+"EVI2diff.tif")


if str(NDVI_checked) == 'true': #If the NDVI box is checked then this if statement will begin.
    NDVI = (NIR-Red)/(NIR+Red)
    NDVI.save(output_test+"NDVI.tif")
```

```
NDVIdiff = PPI - NDVI

NDVIdiff.save(output_test+"NDVIdiff.tif")
```