

# Metodologia de cálculo de anomalias do Índice de Vegetação (NDVI do Satélite Modis) utilizando Arcgis 10 e Python 2.7



Governador do Estado  
Carlos Moisés da Silva

Secretário de Estado da Agricultura e da Pesca  
Ricardo de Gouvêa

Presidente da Epagri  
Edilene Steinwandter

Diretores

Célio Haverroth  
Desenvolvimento Institucional

Giovani Canola Teixeira  
Administração e Finanças

Humberto Bicca Neto  
Extensão Rural e Pesqueira

Vagner Miranda Portes  
Ciência, Tecnologia e Inovação



## **BOLETIM TÉCNICO Nº 195**

# **Metodologia de cálculo de anomalias do Índice de Vegetação (NDVI do Satélite Modis) utilizando Arcgis 10 e Python 2.7**

**Luiz Fernando de Novaes Vianna  
Kleber Trabaquini  
Wilian da Silva Ricce**



**Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina  
Florianópolis  
2020**

Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina (Epagri)  
Rodovia Admar Gonzaga, 1.347, Itacorubi, Caixa Postal 502  
CEP 88034-901, Florianópolis, Santa Catarina, Brasil  
Fone: (48) 3665-5000, fax: (48) 3665-5010  
Site: [www.epagri.sc.gov.br](http://www.epagri.sc.gov.br)

Editado pelo Departamento Estadual de Marketing e Comunicação (DEMC)

Revisor ad hoc: André de Lima – Instituto Nacional de Pesquisas Espaciais/INPE  
Editoração técnica: Paulo Sergio Tagliari  
Revisão textual: Laertes Rebelo  
Diagramação: Vilton Jorge de Souza

*On-line*: junho de 2020

É permitida a reprodução parcial deste trabalho desde que citada a fonte.

### Ficha catalográfica

VIANNA, L.F.N.; TRABAQUINI, K.; RICCE, W.S. **Metodologia de cálculo de anomalias do Índice de Vegetação (NDVI do Satélite Modis) utilizando Arcgis 10 e Python 2.7.** Florianópolis, 2020. 24p. (Epagri. Boletim Técnico, 195).

Processamento de dados de satélite; Sistema de informações geográficas; Plataforma de dados digitais.

ISSN 2674-9513 (*On-line*)



## **Autores**

### **Luiz Fernando de Novaes Vianna, Dr.**

Centro de Informações de Recursos Ambientais  
e de Hidrometeorologia de Santa Catarina – Ciram  
Av. Admar Gonzaga, 1.347, Itacorubi, Caixa postal 502  
CEP 88034-901, Florianópolis, SC, Brasil  
Fone: (48) 3665-5161  
E-mail: vianna@epagri.sc.gov.br

### **Kleber Trabaquini, Dr.**

Centro de Informações de Recursos Ambientais  
e de Hidrometeorologia de Santa Catarina – Ciram  
Av. Admar Gonzaga, 1.347, Itacorubi, Caixa postal 502  
CEP 88034-901, Florianópolis, SC, Brasil  
Fone: (48) 3665-5121  
E-mail: klebertrabaquini@epagri.sc.gov.br

### **Wilian da Silva Ricce, Dr.**

Centro de Informações de Recursos Ambientais  
e de Hidrometeorologia de Santa Catarina – Ciram  
Av. Admar Gonzaga, 1.347, Itacorubi, Caixa postal 502  
CEP 88034-901, Florianópolis, SC, Brasil  
Fone: (48) 3665-5150  
E-mail: wilianricce@epagri.sc.gov.br



## **Apresentação**

Monitorar os estágios de desenvolvimento das lavouras e pomares durante o ano é fundamental para acompanhar o desenvolvimento vegetativo e as fases fenológicas. Além disso, o acompanhamento sazonal das diferentes coberturas vegetais auxilia na compreensão da dinâmica do uso e da cobertura do solo. Uma das formas de realizar essa tarefa em nível estadual é através do cálculo dos Índices de Vegetação (NDVI) e do cálculo das suas anomalias.

Esta publicação tem por objetivo apresentar um método automatizado de cálculo da anomalia do NDVI do satélite Modis (MOD13) e disponibilizar um código em Python para uso no sistema de informações geográficas ArcGis. A obra é destinada a técnicos e pesquisadores e a usuários do ArcGis que necessitam automatizar processos que dependam do cálculo da anomalia do NDVI.

**A Diretoria Executiva**



# Sumário

|                                                                         |           |
|-------------------------------------------------------------------------|-----------|
| <b>Introdução.....</b>                                                  | <b>9</b>  |
| <b>1 Base de dados .....</b>                                            | <b>12</b> |
| <b>2 Metodologia de cálculo da anomalia do índice de vegetação.....</b> | <b>13</b> |
| 2.1 Resumo das etapas.....                                              | 13        |
| 2.1.1 Download dos dados .....                                          | 13        |
| 2.1.2 Organização dos arquivos e pastas (Python script 1) .....         | 13        |
| 2.1.3 Pré-processamento dos dados (Python script 2).....                | 13        |
| 2.1.4 Cálculo da anomalia do NDVI (Python script 3).....                | 14        |
| 2.2 Detalhamento das etapas e scripts comentados.....                   | 14        |
| 2.2.1 Download dos dados .....                                          | 14        |
| 2.2.2 Organização dos arquivos e pastas.....                            | 16        |
| 2.2.3 Scripts .....                                                     | 17        |
| 2.2.4 Linhas de comando.....                                            | 22        |
| <b>Referências.....</b>                                                 | <b>23</b> |



# Introdução

O conhecimento da dinâmica da vegetação em diferentes escalas espaciais e temporais é importante tanto para a gestão ambiental quanto para o planejamento agrícola. Uma das ferramentas disponíveis para acompanhar essa dinâmica de forma rápida e em grandes áreas é o dado de sensoriamento remoto orbital. Os satélites possuem sensores ópticos que captam a luz solar refletida por cada alvo na superfície da Terra e, no caso da vegetação, cada espécie ou grupo de espécies, apresenta um padrão radiométrico que varia durante o ano. Isso permite identificar e monitorar florestas, pomares ou lavouras, pois cada espécie tem características diferenciadas em termos de ciclo, época de cultivo e desenvolvimento do dossel (Figura 1).

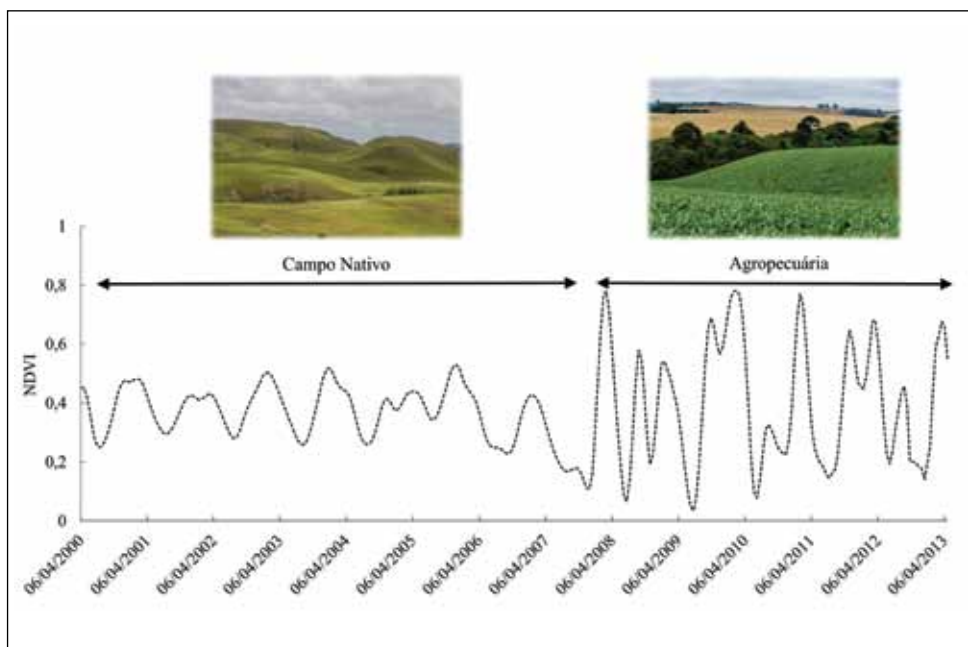


Figura 1. Exemplo de mudança de padrão radiométrico do NDVI em função da cobertura do solo

Desde a década de 1960, os cientistas vêm extraindo e modelando vários parâmetros biofísicos da vegetação com o uso de dados de satélites. Grande parte desse esforço tem envolvido o cálculo de índices de vegetação que podem ser definidos como medidas radiométricas adimensionais que indicam a condição da vegetação-alvo.

Entre os produtos disponíveis obtidos pelos satélites, o Índice de Vegetação por Diferença Normalizada - NDVI (Normalized Difference Vegetation Index) é frequentemente utilizado para identificar e analisar o comportamento da vegetação. Esse índice avalia o estado da vegetação com base na diferença entre a absorção da radiação solar das faixas de comprimento de onda pela planta. Quanto mais ativa a vegetação pela atividade da clorofila, maior é a reflectância da radiação solar na faixa do infravermelho próxima e menor na região do vermelho. Já as plantas com menor atividade fotossintética ou dossel malformado apresentam menor reflectância na região do infravermelho próxima e maior na região do vermelho. Essa diferença permite a classificação da vegetação e o reconhecimento de alguns padrões (Figura 2).

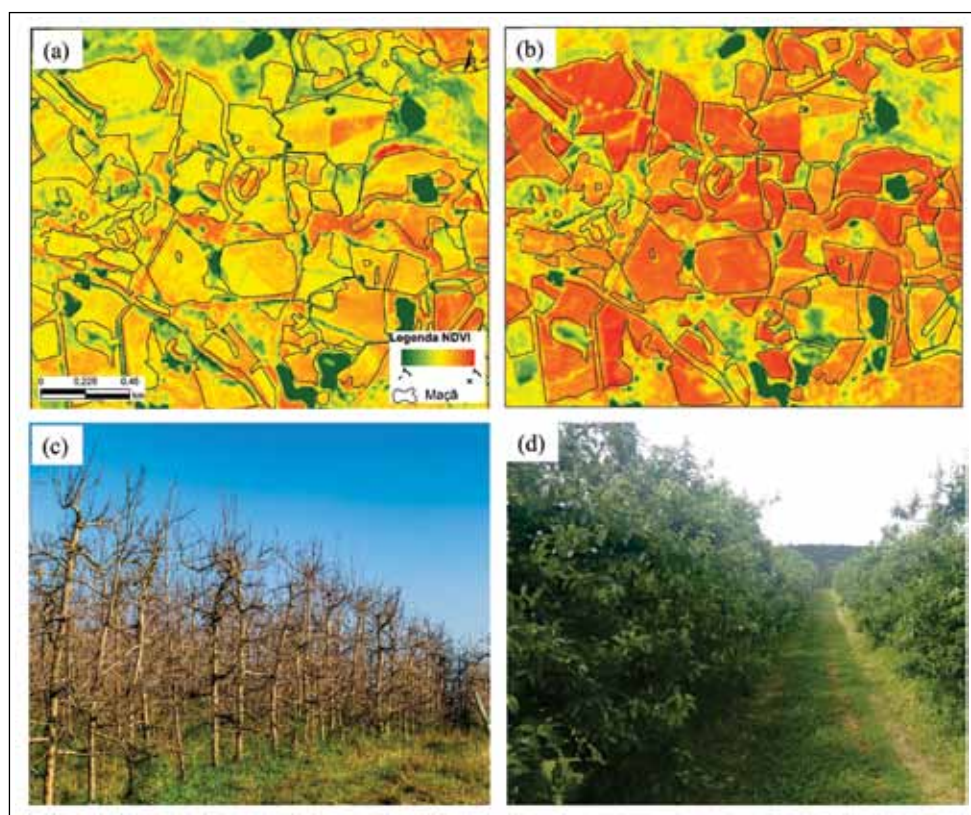


Figura 2. Diferença de padrões de coloração do NDVI em pomares de maçã em fases fenológicas e épocas do ano distintas: (a e b) Imagem NDVI em 14/06/2018 (inverno) com pomares de maçã em época de desfolha e baixo índice foliar; (a e c) imagem NDVI em 11/11/17 (primavera) com pomares em época de brotação maçã com alto índice foliar (b e d)

O sensor MODIS (Moderate Resolution Imaging Spectroradiometer) presente nos satélites Terra e Aqua (<https://modis.gsfc.nasa.gov/>) se destaca pela resolução temporal e espacial do índice de vegetação (Didan, 2015). A composição dos dois satélites permite o monitoramento com imagens compostas a cada 8 dias e resolução espacial de 250m. Com isso, é possível monitorar todo o estado de Santa Catarina com apenas um recorte da imagem global (Figura 3, Tile H13V11).

A série histórica de NDVI do MODIS teve início em 2000, proporcionando em 2019 uma base de dados completa de 19 anos, com 720 arquivos de imagens decendiais e um volume aproximado de 25GB. Para processar esse volume de dados são necessários computadores de alto desempenho e processos automatizados.

O cálculo da anomalia decendial do NDVI é feito a partir da diferença entre o NDVI do atual decêndio e a média histórica do NDVI para aquele decêndio. Realizar esse cálculo exige que, a cada ano, a média histórica seja recalculada, adicionando-se o NDVI do decêndio corrente. A análise da anomalia do NDVI permite que os pesquisadores avaliem, em todo o estado, que áreas estão com o desenvolvimento vegetativo acima, de acordo ou abaixo da média. O NDVI pode ser (confrontado, sobreposto...) integrado a outros mapas (balanço hídrico ou variáveis climáticas) para auxiliar na compreensão do comportamento da cobertura vegetal naquele período, em determinada região.

O desenvolvimento de rotinas automatizadas para processar os dados dos satélites facilita a operação dos técnicos e pesquisadores para fornecer informações como a anomalia do NDVI. A automação do cálculo do NDVI auxilia, por exemplo, na rápida identificação de regiões afetadas por intempéries climáticas, como geadas e estiagens. O cálculo das anomalias permite também acompanhar o desenvolvimento vegetativo das culturas e algumas mudanças de padrões na cobertura do solo, como desmatamento ou corte de florestas plantadas.

Tanto o NDVI quanto a anomalia do NDVI são indicadores comumente utilizados em monitoramento ambiental e acompanhamento agrícola. Quando as rotinas que geralmente envolvem o seu processamento em Sistemas de Informações Geográficas estão disponíveis aos usuários, a rotina de processamento dos dados torna-se mais acessível.

O objetivo desse trabalho é disponibilizar uma rotina através de um conjunto de scripts para calcular a anomalia decendial do NDVI do sensor MODIS no estado de Santa Catarina. Os scripts foram escritos em Python 2.7 para ArcGis 10.x. Para executá-los é necessário ter o ArcGis 10.x e a extensão Spatial Analyst licenciada.

# 1 Base de dados

Através da plataforma *Boku* (<http://ivfl-info.boku.ac.at>), gerenciada pela University of Natural Resources and Life Sciences, as imagens do NDVI são geradas a partir de uma composição de 16 dias, com uma resolução temporal de 8 dias, e disponibilizadas globalmente (Vuolo, et.al., 2012). Os produtos originais do sensor MODIS Level-3 16-day composite, dos satélites Terra (MOD13 series) e Aqua (MYD13 series) são processados através do algoritmo de filtragem Whittaker smoother (Eliers, 2003; Atkinson, et.al., 2012), que remove os dados espúrios e corrige falhas de preenchimento. Através da plataforma, o usuário tem a opção de gerar as imagens do NDVI por períodos de sete ou dez dias para uma ou mais cenas de 10 x 10 graus de área (Figura 3). O estado de Santa Catarina está coberto por uma única cena, representada pelo quadrante H13V11.

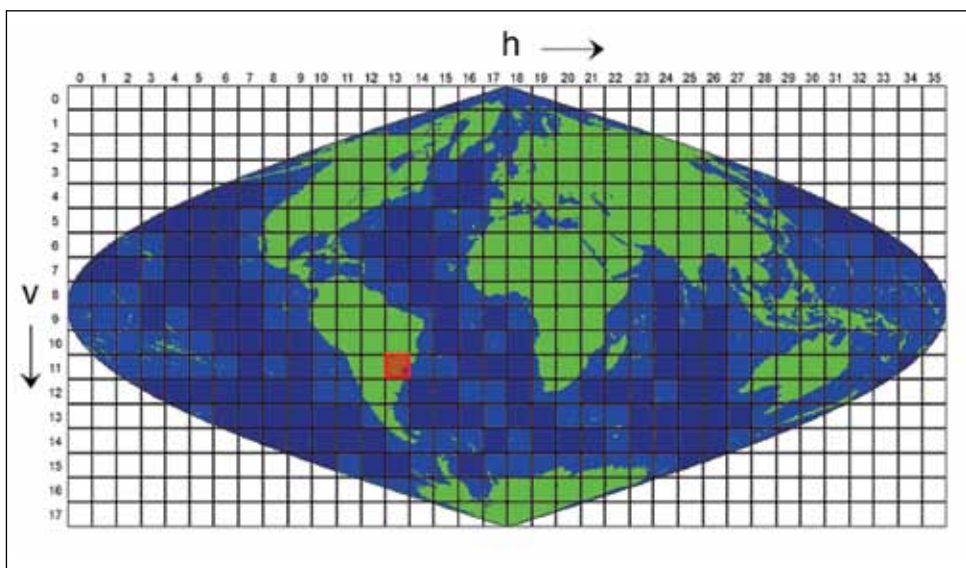


Figura 3. Cobertura global de imagens NDVI e EVI do sensor MODIS e o Tile H13V11 em destaque sobre o estado de Santa Catarina

Adaptado de <https://boku.ac.at/fileadmin/data/H03000/H85000/H85700/images/news/MOD13Q1.005.dailyCheck.png>

## 2 Metodologia de cálculo da anomalia do índice de vegetação

Para calcular a anomalia do NDVI é preciso seguir quatro etapas (Figura 4).

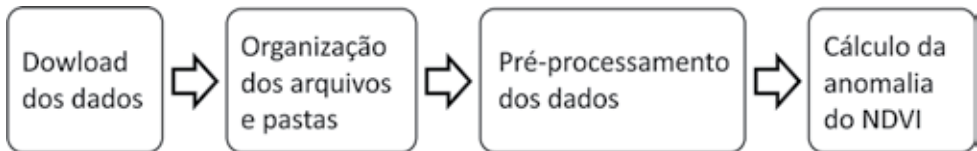


Figura 4. Etapas necessárias para calcular a anomalia do NDVI utilizando os scripts em Python para ArcGis 10.x

### 2.1 Resumo das etapas

#### 2.1.1 Download dos dados

Selecionar e baixar os dados matriciais (Geotif) filtrados do NDVI de <http://ivfl-info.boku.ac.at/satellite-data-processing/dataprocess-global>;

#### 2.1.2 Organização dos arquivos e pastas (Python script 1)

- Armazenar os dados matriciais originais em X:\Modis250\Originals;
- Renomear os dados matriciais originais para o padrão “ndvi\_YYYY\_dec\_dd.tif”, onde “YYYY” é o ano e “dd” é o decêndio;
- Transferir os arquivos renomeados de ...\\Modis250\Originals para ...\\Modis250\NDVI10dias.

#### 2.1.3 Pré-processamento dos dados (Python script 2)

- Projetar os dados matriciais originais de Lat/Lon WGS 1984 para Lat/Lon SIRGAS2000;
- Substituir os valores espúrios de NDVI nos dados matriciais por 0 ou 9999:  $\text{NDVI} = (\text{if NDVI} < 1, 0 \text{ and if NDVI} \geq 9999, 9999)$ ;
- Padronizar os dados matriciais:  $\text{NDVI} = (\text{NDVI} - \text{MEAN}(\text{NDVI})) / \text{STD}(\text{NDVI})$ ;
- Transformar os dados matriciais para uma escala 0 – 100:  $\text{NDVI} = ((\text{NDVI} - \text{MIN}(\text{NDVI})) / (\text{MAX}(\text{NDVI}) - \text{MIN}(\text{NDVI}))) * 100$ ;
- Recortar os dados matriciais padronizados e escalonados utilizando a máscara do estado de Santa Catarina.

### 2.1.4 Cálculo da anomalia do NDVI (Python script 3)

- a) Calcular a média decendial do NDVI: mean  
\_dec\_dd.tif = MEAN (ndvi\_YYY1\_dec\_dd.tif, ndvi\_YYY2\_dec\_dd.tif, ndvi\_YYY3\_dec\_dd.tif,...);
- b) Calcular a anomalia do NDVI para cada ano por decêndio: anom\_yyyy\_dec\_dd.tif = MINUS(ndvi\_YYYY\_dec\_dd.tif, mean\_dec\_dd.tif).

## 2.2 Detalhamento das etapas e scripts comentados

### 2.2.1 Download dos dados

Os dados utilizados para calcular as anomalias do NDVI no estado de Santa Catarina devem ser obtidos no site do projeto BOKU da universidade de Viena (<https://boku.ac.at/rali/ivfl>). O link direto para download é: <http://ivfl-info.boku.ac.at/satellite-data-processing/dataprocess-global>

O leiaute da plataforma está representado nas Figuras 5, 6 e 7, onde são definidos os produtos, a região de interesse, o período dos dados e os parâmetros de filtragem.

**Product:** MODIS NDVI 16-Day Global 250m smooth time series (Figura 5)

**Sensor:** Combined Terra & Aqua

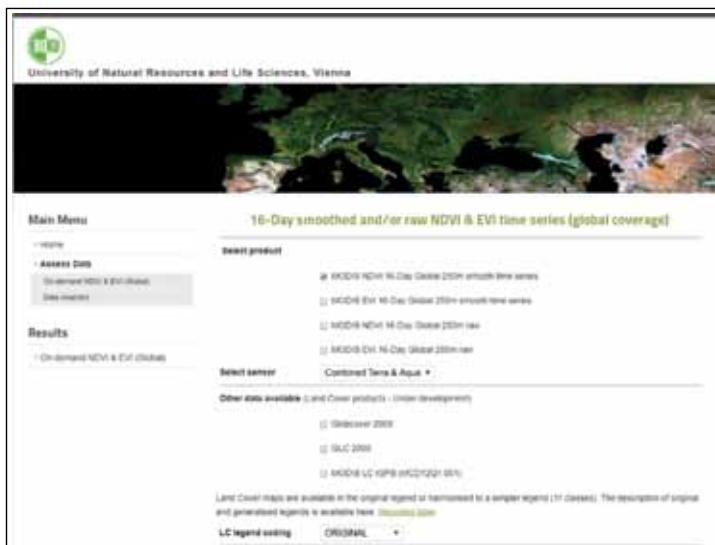


Figura 5. Captura de tela do sítio da plataforma BOKU para seleção dos produtos para download (Acesso em 05/02/2020)

**Region of interest:** ULCL Lat: -25; ULCL Lng: -54; LRC Lat: -30; LRC Lng: -48  
(Figura 6)

Select region of interest (Click and drag the the Left corner (L) and the Right corner (R) to re-size the rectangle

ULCL Lat: -25 ULCL Lng: -54 LRC Lat: -30 LRC Lng: -48

The maximum processable area is 10 Million pixels. Please, contact us if you require a larger image extent. The current selection is:

Date volume (in Megabytes) per selected product will be:

Estimated filtering time (in minutes) per selected product (H2O1 or EVI) is:

Figura 6. Captura de tela do sítio da plataforma BOKU para seleção da área de interesse para download dos produtos (Acesso em 05/02/2020)

**Filtering parameters:** Default (Figura 7)

**File output parameters:** Temporal resolution: 10-days; File type: GeoTIFF (.tif); **Projection (EPSG code):** 4326 – Lat/Lon WGS 1984

Select dates of start and end

Please note that for short (< 1 year) time series smoothing, we will process and filter at least 1 year data.

Start date: 2000-01-01

End date: 2000-01-01

Filtering parameters (if unsure, leave the default values)

Lambda: 0000

Number of iterations: 2

File output parameters (if unsure, leave the default values)

Temporal resolution: 10-days

File type: GeoTIFF (.tif)

Projection (EPSG code): 4326

e-mail: [input field]

Job name: [input field]

Enter the code: [input field] Lfm rZ

Submit

Figura 7. Captura de tela do sítio da plataforma BOKU para configuração do período de aquisição dos dados, dos parâmetros de filtragem, da resolução temporal e do formato dos arquivos (Acesso em 05/02/2020)

## 2.2.2 Organização dos arquivos e pastas

Os arquivos baixados devem ser armazenados na pasta “X:\Modis250\Originais”.

Os arquivos e pastas devem seguir um padrão de nomenclatura. Todas as pastas abaixo devem ser criadas no computador antes de rodar os scripts. Em cada script a letra “X:” deve ser substituída pela letra que representa o drive no qual os arquivos e pastas ficarão armazenados.

X:\Modis250 – Pasta principal de trabalho (Workspace)

X:\Modis250\Originais – Pasta de download dos arquivos originais. Nesta pasta ficam apenas os arquivos mais recentes, conforme os dados forem sendo baixados do site periodicamente.

X:\Modis250\Originais\_bkp – Pasta onde os arquivos originais serão armazenados e guardados após a padronização dos nomes.

X:\Modis250\NDVI\_10dias – Pasta para onde serão transferidos os arquivos originais renomeados para o padrão de processamento (resultados do Python script 1)

X:\Modis250\NDVI\_10dias\_norm – Pasta dos arquivos pré-processados. Arquivos projetados para Lat/Lon SIRGAS 2000, padronizados, escalonados para 0-100 e recortados para a área de interesse (máscara do estado de Santa Catarina) (resultados do Python script 2)

X:\Modis250\NDVI\_10dias\_mean – Pasta dos arquivos resultantes do cálculo do NDVI médio para cada decêndio (resultados do Python script 3)

X:\Modis250\NDVI\_10dias\_anom – Pasta dos arquivos resultantes do cálculo de anomalia do NDVI para cada decêndio (resultados do Python script 3)

X:\Python – Pasta de armazenamento dos Python scripts

X:\Temp – Pasta temporária (para os arquivos temporários gerados pelo Python script 2)

X:\SRTM – Pasta onde fica armazenado o arquivo de máscara do estado de Santa Catarina.

**P.S.: Antes de rodar os scripts, o idioma do MS Windows deverá ser configurado para inglês (EN)**

## 2.2.3 Scripts

### Python script 1

- Renomeia os arquivos matriciais originais do padrão do BOKU (MODIS.ndvi.YYYYjjj.yL6000.BOKU.tif) para o padrão de processamento “ndvi\_YYYY\_dec\_dd.tif”, onde “YYYY” é o ano, “jjj” é o dia Juliano and “dd” é o decêndio.

- Transfere os arquivos renomeados de X:\Modis250\Originais para X:\Modis250\NDVI10dias. Isso mantém a pasta de downloads sempre vazia e permite o reproprocessamento apenas dos arquivos mais recentes.

```
def main():
    import arcpy
    import os
    import shutil

    arcpy.env.overwriteOutput = True

    ws_in = r'X:\Modis250\Originais'
    ws_out = r'X:\Modis250\NDVI_10dias'

    arcpy.env.workspace = ws_in
    lst_ras = arcpy.ListRasters("*.tif")

    for ras_name in sorted(lst_ras):

        in_name_path = os.path.join(ws_in, ras_name)
        y = ras_name[11:15]
        j = int(ras_name[15:18])
        d = GetDecadeFromJulian(int(y), j)

        out_name = "ndvi_{}_dec_{}.tif".format(y, d)
        out_name_path = os.path.join(ws_out, out_name)
        shutil.move(in_name_path, out_name_path)
        print(out_name + " renamed")

    def GetDecadeFromJulian(year, j):
        import datetime
        date = datetime.datetime(year-1, 12, 31) + datetime.timedelta(j)
        m = (date.month - 1) * 3
```

```

if date.day <= 10:
    d = 1
elif date.day <= 20:
    d = 2
else:
    d = 3
return m + d

if __name__ == '__main__':
    main()

```

## Python script 2

- Projeta os dados matriciais renomeados ("ndvi\_YYYY\_dec\_dd.tif") de Lat/Lon WGS 1984 para Lat/Lon SIRGAS2000 utilizando a transformação geográfica "SAD\_1969\_To\_WGS\_1984\_14 + SAD\_1969\_To\_SIRGAS\_2000\_1".
- Substitui os valores espúrios menores que zero e maiores que 9999: NDVI = Con("%NDVI%"<1,0,Con("%NDVI%">9998,9999,"%NDVI%")).
- Padroniza os dados matriciais: NDVI = (NDVI – Mean(NDVI))/Std(NDVI).
- Transforma os dados matriciais para a escala 0 – 100: NDVI = ((NDVI-Min(NDVI))/(Max(NDVI)-Min(NDVI)) \* 100.
- Recorta os dados matriciais na área de interesse utilizando a máscara do estado de Santa Catarina (X:\Modis250\SRM\mask.tif)
- Grava os arquivos finais em X:\Modis250\NDVI\_10dias\_norm

```

def main():
    import arcpy
    import os
    arcpy.CheckOutExtension("spatial")
    arcpy.env.overwriteOutput = True

```

```

# Definindo o ambiente do ArcGis
ws_scr = r'X:\Modis250\Temp' #pasta temporária
ws_in = r'X:\Modis250\NDVI_10dias' #Pasta dos dados de entrada
ws_out = r'X:\Modis250\NDVI_10dias_norm' #Pasta dos dados de saída
mask = r'X:\Modis250\SRM\mde_sc_90.tif' #Pasta do raster que será
utilizado como máscara para recorte da área de interesse
arcpy.env.scratchWorkspace = ws_scr
arcpy.env.workspace = ws_in
lst_ras = arcpy.ListRasters("*.tif")
lst_ras.sort()

```

```

# Pré-processando os rasters originais
for raster1 in lst_ras:
    print ("Processing: " + raster1)

    #outRaster1= Conversão do raster original projetado em CGS-WGS1964
    para CGS-SIRGAS-2000
    outRaster1 = os.path.join(ws_scr, "rep_{}".format(raster1)) # Arquivo
    temporário
    arcpy.ProjectRaster_management(raster1, outRaster1,"GEOGCS['GCS_
    SIRGAS_2000',DATUM['D_SIRGAS_2000',SPHEROID['GRS_1980',6378137.0,298.
    257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]]",
    "NEAREST", "1.8000795216263E-03 1.80029836029412E-03", "'SAD_1969_
    To_WGS_1984_14 + SAD_1969_To_SIRGAS_2000_1'", "", "GEOGCS['GCS_
    WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.2572
    23563]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]]")
    print ("rep_{}".format(raster1) + " projected")

    #outRaster2 = outRaster1 recortado pela área de interesse (máscara)
    outRaster2 = arcpy.sa.ExtractByMask(outRaster1, mask) # Arquivo
    temporário
    outRaster2.save(os.path.join(ws_scr, "mask_{}".format(raster1)))
    print ("mask_{}".format(raster1) + " masked")

    #outRaster3 = Substituição dos valores espúrios do outRaster2
    outRaster3 = arcpy.sa.Con(outRaster2 < 1, 0, arcpy.sa.Con(outRaster2 >
    9998, 9999, outRaster2)) # Arquivo temporário
    outRaster3.save(os.path.join(ws_scr, "aju_{}".format(raster1)))
    print ("aju_{}".format(raster1) + " adjusted")

    #outRaster4 = Padronização do outRaster3
    ras_mean = float(arcpy.GetRasterProperties_management(outRaster3,
    "MEAN").getOutput(0))
    ras_std = float(arcpy.GetRasterProperties_management(outRaster3,
    "STD").getOutput(0))
    outRaster4 = ((outRaster3-ras_mean)/ras_std) # Temporary file
    outRaster4.save(os.path.join(ws_scr, "ras_std_{}".format(raster1)))
    print ("ras_std_{}".format(raster1) + " standardized")

    #outRaster = Raster final na escala 0-100
    ras_max = float(arcpy.GetRasterProperties_management(outRaster4,

```

```

"MAXIMUM").getOutput(0))
    ras_min = ras_min = float(arcpy.GetRasterProperties_
management(outRaster4, "MINIMUM").getOutput(0))
    outRaster = ((outRaster4-ras_min)/(ras_max-ras_min) * 100) # Final result
    outRaster.save(os.path.join(ws_out, "norm_{}".format(raster1)))
    print ("norm_{}".format(raster1) + " processed")

arcpy.CheckInExtension("spatial")

if __name__ == '__main__':
    main()

```

### Python script 3

- Calcula o NDVI médio para cada decêndio: mean\_dec\_dd.tif = MEAN (ndvi\_YYY1\_dec\_dd.tif, ndvi\_YYY2\_dec\_dd.tif, ndvi\_YYY3\_dec\_dd.tif,...)
- Calcula a anomalia do NDVI para cada decêndio: anom\_yyyy\_dec\_dd.tif = ndvi\_YYYY\_dec\_dd.tif - mean\_dec\_dd.tif

```

def main():
    import arcpy
    import os
    arcpy.env.overwriteOutput = True

    # CCheca a licença do Spatial Analyst
    arcpy.CheckOutExtension("spatial")

    # Define as pastas de trabalho
    ws_norm = r'X:\Modis250\NDVI_10dias_norm'
    ws_mean = r'X:\Modis250\NDVI_10dias_mean'
    ws_anom = r'X:\Modis250\NDVI_10dias_anom'

    # cria uma lista de todos os rasters normalizados
    arcpy.env.workspace = ws_norm
    lst_ndvi_ras = arcpy.ListRasters()
    #print ("Raster list: {}".format(lst_ndvi_ras))
    # cria uma lista de anos
    decades = GetListOfDecades(lst_ndvi_ras)
    decades.sort()
    #print ("Decades list: {}".format(decades))

```

```

# Roda para cada ano
for decade in decades:
# Separa os rasters por ano
lst_ndvi_decade = GetListOfRasterForGivenDecade(lst_ndvi_ras, decade)
print("Decade by year list: {}".format(lst_ndvi_decade))

# Calcula a média do NDVI por decêndio
lst_ndvi_decade_paths = [os.path.join(ws_norm, r) for r in lst_ndvi_decade]
#print("lst_ndvi_decade_paths: {}".format(lst_ndvi_decade_paths))
mean_ras = arcpy.sa.CellStatistics(lst_ndvi_decade_paths, "MEAN",
"DATA")

# Grava o raster médio de cada decêndio
# mean_dec_d.tif
out_name_mean = "mean_dec_{}.tif".format(decade) ###
out_name_path_mean = os.path.join(ws_mean, out_name_mean)
print("Mean raster: {}".format(out_name_mean))
mean_ras.save(out_name_path_mean)

# Roda para cada decêndio
for ndvi_ras in lst_ndvi_decade:
# calcula a anomalia do NDVI para cada raster descendial
# anom_2002_dec_1.tif = norm_ndvi_2002_dec_1.tif - mean_dec_1.tif;
print("Processing: {}".format(ndvi_ras))
# minus_ras = arcpy.sa.Minus(ndvi_ras, out_name_path_mean)
out_name_path_norm = os.path.join(ws_norm, ndvi_ras)
minus_ras = arcpy.Raster(out_name_path_norm) - arcpy.Raster(out_
name_path_mean)

# Grava o raster da anomalia
out_name_anom = ndvi_ras.replace("norm_ndvi", "anom")
out_name_path_anom = os.path.join(ws_anom, out_name_anom)
minus_ras.save(out_name_path_anom)
print("Anom: {}".format(out_name_path_anom))

# return SA license
arcpy.CheckInExtension("spatial")
def GetListOfDecades(lst):
# norm_ndvi_yyyy_dec_d.tif
lst1 = list(set([r.split("_")[4] for r in lst]))
return [r.split("_")[0] for r in lst1]

```

```
def GetListOfRasterForGivenDecade(lst, decade):
    lst1 = []
    search = "{}.tif".format(decade)
    for r in lst:
        if search in r.lower():
            lst1.append(r)
    return lst1

if __name__ == '__main__':
    main()
```

## 2.2.4 Linhas de comando

A seguir temos as linhas de comando para rodar os scripts em sequência através da linha de comando (ndvi.bat). Substituir “X:” pela letra correspondente ao drive onde estão os arquivos e pastas. Substituir “V” pela versão do ArcGis 10, onde está instalado o Python 2.7.

```
@ECHO OFF
REM Runs NDVI anomaly scripts
ECHO Backup original files
xcopy X:\Modis250\Originais\*.tif X:\Modis250\Originais_bkp /c /i /y /z
PAUSE
ECHO Selecting decade and renaming original files to “ndvi_YYYY_dec_
dd.tif” standard.
C:\Python27\ArcGIS10.V\Python.exe X:\Modis250\Python\Script1.py
ECHO NDVI files renamed, hit enter to pre-processing data.
PAUSE
ECHO Projecting, removing spurious data, standardizing and scaling ndvi
decade files to 0-100.
C:\Python27\ArcGIS10.V\Python.exe X:\Modis250\Python\Script2.py
ECHO NDVI pre-processed, hit enter to calculate anomalies.
PAUSE
ECHO Calculating decade anomalies.
C:\Python27\ArcGIS10.V\Python.exe X:\Modis250\Python\Script3.py
Del X:\Modis250\Temp\ /s /q
Del X:\Modis250\NDVI_10dias\ /s /q
ECHO Done.
PAUSE
```

## Referências

ATKINSON, P. M., Jeganathan, C., Dash, J., Atzberger, C.. Inter-comparison of four models for smoothing satellite sensor time-series data to estimate vegetation phenology. **Remote Sensing of Environment**, v.123, pp.400–417.

DIDAN, K.. **MOD13Q1 MODIS/Terra Vegetation Indices 16-Day L3 Global 250m SIN Grid V006**. 2015, distributed by NASA EOSDIS Land Processes DAAC, <https://doi.org/10.5067/MODIS/MOD13Q1.006>. Acesso em 05/02/2020.

EILERS, P. H. C. (2003). A perfect smoother. **Analytical Chemistry**, v.75, n.14, pp.3631–3636, 2003.

JÖNSSON, P., Eklundh, L. (2002).. Seasonality extraction by function fitting to time-series of satellite sensor data. **IEEE Transactions on Geoscience and Remote Sensing**, 40(8), p.1824–1832.

VUOLO, F., MATTIUZZI, M., KLISCH, A., ATZBERGER, C.. (2012). Data service platform for MODIS NDVI time series pre-processing at BOKU Vienna: current status and future perspectives. **Proc. SPIE Earth Resour. Environ. Remote Sens. /GIS Appl. III**. 8538. 10.1117/12.974857.





[www.epagri.sc.gov.br](http://www.epagri.sc.gov.br)



[www.youtube.com/epagritv](http://www.youtube.com/epagritv)



[www.facebook.com/epagri](http://www.facebook.com/epagri)



[www.twitter.com/epagrioфициal](http://www.twitter.com/epagrioфициal)



[www.instagram.com/epagri](http://www.instagram.com/epagri)



[linkedin.com/company/epagri](http://linkedin.com/company/epagri)



<http://publicacoes.epagri.sc.gov.br>