

LABORATÓRIOS DIDÁTICOS DE GEOPROCESSAMENTO

Sensoriamento Remoto

Estatísticas, histogramas, contraste e composições de imagens
no GEE

Docente: Prof.^a Dr.^a Mariana Abrantes Giannotti

Elaboração do roteiro: Leonardo Alves Godoy

Revisão do roteiro: Christian de Nazareth Teixeira

Objetivos

- Obtendo os valores de um *pixel* no mapa no GEE
- Estatística dos dados de uma imagem no GEE
- Escalonando valores da imagem
- Gerando histogramas no GEE
- Ajustando o contraste de uma imagem no GEE
- Composição colorida com ajuste de contraste simples no GEE
- Composição colorida com ajuste de contraste individuais no GEE

1 Introdução

Neste laboratório serão apresentadas as operações para calcular estatísticas dos valores dos *pixels* nas bandas de um *raster* e gerar gráficos de suas distribuições. Também será visto a forma mais simples de se extrair valores de um *pixel* em uma imagem no mapa do GEE. Além disso, serão introduzidas formas para se ajustar o contraste de uma imagem adicionada ao mapa e como formar composições coloridas com as bandas dos dados sendo trabalhados.

Cabe ressaltar que neste laboratório os blocos de código serão disponibilizados no roteiro para se efetuar a cópia para o *script* que será construído, mas deve-se ler atentamente as instruções e as linhas de código para compreender o que está sendo feito.

2 Referências teóricas

Os fundamentos teóricos básicos para este laboratório são derivados dos apresentados no laboratório anterior, cobrindo o armazenamento de dados em arquivos *raster*, mais ferramentas de estatística e análise exploratória de dados comuns. Além desses, é possível encontrar em CRÓSTA (1992) os conceitos relativos ao ajuste de contraste em composições coloridas com as bandas de imagens. Na parte de programação, durante o laboratório, serão oportunamente

apresentados os conceitos necessários para o desenvolvimento das aplicações de interesse.

3 Roteiro prático

3.1 Preparação dos dados no GEE

De forma similar ao laboratório anterior, deve-se criar e salvar, em local apropriado, um novo *script*.

Um ponto único, em uma *feature*, deve ser criado e atribuído a uma variável localizada na cidade de São Paulo - nomear como **“saoPaulo”**. Esse ponto será utilizado como referência durante o laboratório. Além desse ponto, deve-se criar outra *feature* sobre a cidade de São Paulo com um retângulo com nome **“retanguloSelecao”**, onde achar apropriado, pois ele será utilizado para recortar parte da imagem para agilizar o processamento dos exemplos.

Também deve-se importar os dados de uma coleção em uma variável. Nos exemplos preparados para este laboratório, a coleção utilizada foi a **“Sentinel-2 MSI: MultiSpectral Instrument, Level-1C”**, deve-se nomear a coleção como **“colecacaoSentinel2_1C”**. Deve-se ler atentamente a descrição da coleção, as informações de suas bandas e, também, observar os nomes das propriedades das imagens e seus detalhes. Nessa coleção, o termo **“Level-1C”** determina o tipo do produto disponibilizado pela fornecedora das imagens - a agência espacial européia, conhecida pela sigla *ESA*. O produto **“Level-1C”** possui em seus *pixels* o valor reflectância no topo da atmosfera - *Top Of Atmosphere (TOA)* - multiplicado por 10.000. De forma similar, é distribuído o produto **“Level-2A”**, que apresenta a reflectância da superfície - *Bottom Of Atmosphere (BOA)* - multiplicada por 10.000. O **“Level-2A”** é obtido através da chamada correção atmosférica dos dados do **“Level-1C”**¹.

Ao fim das importações de dados acima, o setor **“Imports”** do GEE deve ficar similar ao apresentado na Figura 1.

¹ Detalhes sobre o processamento dos dados do *Sentinel-2* podem ser obtidos em https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook

Observação: Fazer a seleção de um polígono retangular pequeno na área de São Paulo para que o processamento fique mais rápido.

Figura 1 - Seção “Imports” após a importação dos dados

```
Imports (4 entries)
▶ var colecaoSentinel2_1C: ImageCollection "Sentinel-2 MSI: MultiSpectral Instrument, Level-1C"
▶ var saoPaulo: Point (-46.65, -23.56)
▶ var retanguloSelecao: Polygon, 4 vertices
```

Para o restante da prática, uma única imagem será selecionada para uso. Na Figura 2, observa-se como extrair a primeira imagem da coleção seguindo alguns requisitos, de forma similar ao visto nos laboratórios anteriores. Primeiro, “**filterBounds()**” separa as imagens da coleção que coincidem com o ponto “**saoPaulo**”. Na sequência, através dos metadados, são selecionadas apenas imagens com menos de 20% de cobertura de nuvens, prosseguindo com a seleção através de um período de tempo. Finalmente, o método “**first()**” é invocado para extrair a primeira imagem da coleção resultante. Na imagem retornada, são selecionadas apenas as bandas “**B2**”, “**B3**”, “**B4**” e “**B8**”. O método “**clip()**” recorta a porção da imagem no retângulo criado.

Figura 2 - Seleção de imagem para ser trabalhada

```
1 //Seleção da imagem a ser trabalhada
2 var imgHistograma = colecaoSentinel2_1C.filterBounds(saoPaulo)
3                                     .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 20)
4                                     .filterDate('2019-01-01', '2020-01-01')
5                                     .first()
6                                     .select('B[2-4]', 'B8')
7                                     .clip(retanguloSelecao);
```

```
//Seleção da imagem a ser trabalhada.
var imgHistograma = colecaoSentinel2_1C.filterBounds(saoPaulo)
                                     .filterMetadata('CLOUDY_PIXEL_PERCENTAGE',
                                     'less_than', 20)
                                     .filterDate('2019-01-01', '2020-01-01')
                                     .first()
                                     .select('B[2-4]', 'B8')
                                     .clip(retanguloSelecao);
```

3.2 Exibindo valores de um *pixel* no mapa no GEE

Para esta parte do laboratório, primeiramente deve ser adicionada uma banda da imagem ao mapa (imagem monocromática). Na Figura 3 está um exemplo de como se efetuar esse processo. O segundo parâmetro da função, que ajusta a

visualização, segue vazio para manter os valores do padrão. O terceiro parâmetro é apenas um título mais adequado para a camada que é criada no mapa pela função.

Figura 3 - Código para adicionar a banda “B2” ao mapa

```
8 //Adicionando banda "B2" ao mapa
9 Map.addLayer(imgHistograma.select('B2'), "", "Recorte São Paulo - Banda B2");

//Adicionando banda "B2" ao mapa
Map.addLayer(imgHistograma.select('B2'), "", "Recorte São Paulo - Banda B2");
```

A camada adicionada será como a da Figura 4. Embora ela apresente esse aspecto, as informações estão lá. Para conferir, basta clicar na aba “**Inspector**”, no lado superior direito da interface do GEE e em seguida clicar em algum ponto da imagem adicionada no mapa. Na aba “**Inspector**” será apresentado o valor da banda “**B2**” no ponto selecionado - Figura 5. Caso houvesse mais bandas na imagem adicionada, elas teriam seus valores apresentados na aba “**Inspector**”.

Figura 4 - Banda “B2” no mapa com parâmetros de visualização padrões

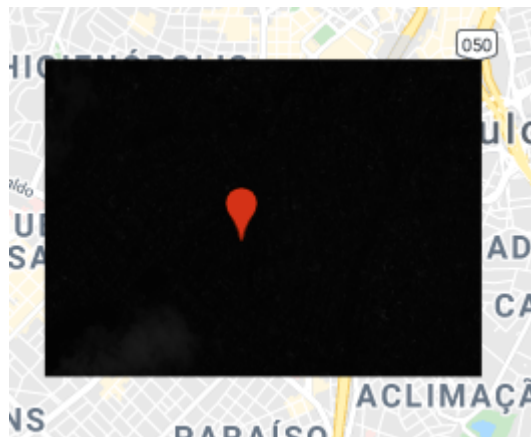


Figura 5 - Aba “**Inspector**” exibindo o valor da reflectância de um ponto

```
Inspector Console Tasks
> Point (-46.63615, -23.54976) at 19m/px
▼ Pixels
  ▼ Recorte São Paulo - Banda B2: Image (1 band)
    B2: 1576
```

3.3 Estatísticas dos dados de uma imagem no GEE

No primeiro laboratório foi visto como utilizar o *reducer* “**mean()**” para calcular a média dos *pixels* de uma região em uma imagem. De forma similar, na Figura 6, é apresentado como se calcula o desvio padrão da distribuição dos valores de uma região em uma imagem. No exemplo apresentado, o parâmetro “**geometry**” não é fornecido. Esse parâmetro determina a região da imagem onde o *reducer* será aplicado. Quando não se determina o parâmetro “**geometry**”, toda a região ocupada pela banda é utilizada como referência. O parâmetro “**maxPixels**” é utilizado pois há uma limitação no número de *pixels* que pode ser processado pelo GEE, esse valor é fornecido, no exemplo, em notação científica (no caso 10 elevado a nona potência). A função “**print()**” é utilizada para visualizar o resultado no “**Console**”. A Figura 6 ainda apresenta outras estatísticas que podem ser obtidas, como a média - *reducer* “**mean()**” - e a variância - *reducer* “**variance()**”.

Figura 6 - Estatísticas da imagem

```
11 //Redutor para calcular o desvio padrão dos valores dos pixels de uma imagem.
12 var dpPixelsBandas = imgHistograma.reduceRegion({
13   reducer: ee.Reducer.stdDev(),
14   maxPixels: 1e9
15 });
16 //Imprime desvio padrão da distribuição dos pixels no console.
17 print("Desvio Padrão: ",dpPixelsBandas);
18
19 //Redutor para calcular a média dos valores dos pixels de uma imagem.
20 var mediaPixelsBandas = imgHistograma.reduceRegion({
21   reducer: ee.Reducer.mean(),
22   maxPixels: 1e9
23 });
24 //Imprime a média da distribuição dos pixels no console.
25 print("Média: ",mediaPixelsBandas);
26
27 //Redutor para calcular a variância dos valores dos pixels de uma imagem.
28 var varianciaPixelsBandas = imgHistograma.reduceRegion({
29   reducer: ee.Reducer.variance(),
30   maxPixels: 1e9
31 });
32 //Imprime a variância da distribuição dos pixels no console.
33 print("Variância: ",varianciaPixelsBandas);

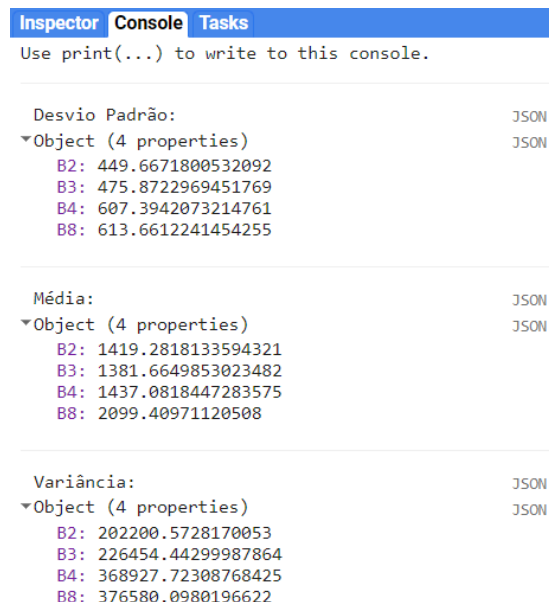
//Redutor para calcular o desvio padrão dos valores dos pixels de uma imagem.
var dpPixelsBandas = imgHistograma.reduceRegion({
  reducer: ee.Reducer.stdDev(),
  maxPixels: 1e9
});
//Imprime desvio padrão da distribuição dos pixels no console.
print("Desvio Padrão: ",dpPixelsBandas);
```

```
//Redutor para calcular a média dos valores dos pixels de uma imagem.
var mediaPixelsBandas = imgHistograma.reduceRegion({
  reducer: ee.Reducer.mean(),
  maxPixels: 1e9
});
//Imprime a média da distribuição dos pixels no console.
print("Média: ",mediaPixelsBandas);

//Redutor para calcular a variância dos valores dos pixels de uma imagem.
var varianciaPixelsBandas = imgHistograma.reduceRegion({
  reducer: ee.Reducer.variance(),
  maxPixels: 1e9
});
//Imprime a variância da distribuição dos pixels no console.
print("Variância: ",varianciaPixelsBandas);
```

Para verificar os valores obtidos, basta ir na aba **“Console”** na parte superior direita e aparecerá algo como na Figura 7 - os valores serão diferentes, pois variam de acordo com o recorte feito.

Figura 7 - Console com estatísticas impressas



3.4 Escalonando os valores da imagem

Como mencionado anteriormente, os valores das coleções do *Sentinel-2* disponíveis no GEE são os valores da reflectância multiplicados por 10.000. Para se

retornar os valores de reflectância, que são uma proporção e por isso estão no intervalo entre 0 e 1, basta dividir o valor de cada pixel da imagem original por 10.000. O código da Figura 8 demonstra como fazer essa divisão, utilizando o método “*divide()*” da imagem que deve ter os valores divididos e passando como parâmetro para o método uma imagem criada em tempo de execução onde todos os pixels são o mesmo valor 10.000.

Figura 8 - Divisão de imagens

```
35 //Divide o valor dos pixels de todas as bandas da imagem por 10mil.  
36 var imgHistogramaNaoEscalonada = imgHistograma.divide(ee.Image.constant(10000));  
37  
//Divide o valor dos pixels de todas as bandas da imagem por 10mil.  
var imgHistogramaNaoEscalonada = imgHistograma.divide(ee.Image.constant(10000));
```

3.5 Gerando Histogramas no GEE

Os valores fornecidos nos produtos do *Sentinel-2* são baseados na reflectância. Para as coleções derivadas do *Landsat*, estão disponíveis nas chamadas *Raw*, os valores *Digital Number (DN)*, que são os valores das intensidades, ou nível de cinza, dos *pixels* das imagens digitais geradas pelo satélite, a partir da onde são calculadas a radiância e as reflectâncias no topo da atmosfera e da superfície. O intervalo dos valores dos *DN's* é derivado da quantidade de *bits* utilizada para armazenar um *pixel*, no caso de uma imagem onde o *DN* está armazenado em 8 *bits*, os níveis de cinza de uma banda ficam entre 0 e 255 ($2^8 - 1$). Essas informações podem impactar no modo como se trabalha com os parâmetros do histograma.

Além de se construir o histograma, é interessante calcular as estatísticas da nova imagem, de forma similar ao visto na Figura 7. Assim, devem ser utilizadas novamente as funções “*mean()*”, “*stdDev()*” e “*variance()*” para, respectivamente, obter o valor médio dos *pixels*, o desvio padrão e a variância da distribuição dos *pixels*. Esse trecho de código deve ficar como na Figura 9, com os valores de reflectância.

Figura 9 - Estatísticas da imagem com os valores de reflectância


```
38 //Redutor para calcular o desvio padrão dos valores dos pixels da imagem não escalonada.
39 var dpPixelsBandas = imgHistogramaNaoEscalonada.reduceRegion({
40   reducer: ee.Reducer.stdDev(),
41   maxPixels: 1e9
42 });
43 //Imprime desvio padrão da distribuição dos pixels no console.
44 print("Desvio Padrão Imagem Não Escalonada: ",dpPixelsBandas);
45
46 //Redutor para calcular a média dos valores dos pixels da imagem não escalonada.
47 var mediaPixelsBandas = imgHistogramaNaoEscalonada.reduceRegion({
48   reducer: ee.Reducer.mean(),
49   maxPixels: 1e9
50 });
51 //Imprime a média da distribuição dos pixels no console.
52 print("Média Imagem Não Escalonada: ",mediaPixelsBandas);
53
54 //Redutor para calcular a variância dos valores dos pixels da imagem não escalonada.
55 var varianciaPixelsBandas = imgHistogramaNaoEscalonada.reduceRegion({
56   reducer: ee.Reducer.variance(),
57   maxPixels: 1e9
58 });
59 //Imprime a variância da distribuição dos pixels no console.
60 print("Variância Imagem Não Escalonada: ",varianciaPixelsBandas);
```

```
//Redutor para calcular o desvio padrão dos valores dos pixels da imagem não escalonada.
```

```
var dpPixelsBandas = imgHistogramaNaoEscalonada.reduceRegion({
  reducer: ee.Reducer.stdDev(),
  maxPixels: 1e9
});
```

```
//Imprime desvio padrão da distribuição dos pixels no console.
print("Desvio Padrão Imagem Não Escalonada: ",dpPixelsBandas);
```

```
//Redutor para calcular a média dos valores dos pixels da imagem não escalonada.
```

```
var mediaPixelsBandas = imgHistogramaNaoEscalonada.reduceRegion({
  reducer: ee.Reducer.mean(),
  maxPixels: 1e9
});
```

```
//Imprime a média da distribuição dos pixels no console.
print("Média Imagem Não Escalonada: ",mediaPixelsBandas);
```

```
//Redutor para calcular a variância dos valores dos pixels da imagem não escalonada.
```

```
var varianciaPixelsBandas = imgHistogramaNaoEscalonada.reduceRegion({
  reducer: ee.Reducer.variance(),
  maxPixels: 1e9
});
```

```
//Imprime a variância da distribuição dos pixels no console.
print("Variância Imagem Não Escalonada: ",varianciaPixelsBandas);
```

O código da Figura 10 mostra como se criar um histograma para uma banda no GEE, basta copiar o código e adequá-lo às variáveis utilizadas no *script*. A

primeira coisa a se notar é o uso da função ***“print()”*** para se exibir o objeto do tipo ***“ui.Chart”*** retornado pelo método ***“ui.Chart.image.histogram()”***. Os parâmetros de ***“ui.Chart.image.histogram()”*** foram passados como um dicionário de dados para facilitar a identificação dos mesmos. Dentre os parâmetros, ***“image”*** é obrigatório, sendo ele a imagem que se deseja analisar. O parâmetro ***“scale”***, não utilizado no exemplo, é a escala do *pixel* para se aplicar o *reducer* que gera o histograma, o padrão é a resolução espacial da banda. Pode ser utilizado caso se queira calcular o histograma em outra escala de resolução menos detalhada, de forma que os valores dos *pixels* são agregados apropriadamente². No exemplo, o parâmetro ***“maxBuckets”*** está comentado, apenas para demonstração, porém convém testá-lo. É um parâmetro opcional que limita o número máximo de barras que podem ser utilizadas para se construir o histograma. Esse valor deve ser uma potência de 2, caso não seja, será arredondado para a potência de 2 mais próxima. ***“minBucketWidth”*** representa a largura mínima das barras que serão geradas e é um parâmetro opcional que caso não seja fornecido, o GEE tenta ajustar uma potência de 2 como valor. Cabe adicionar, que há um número máximo de *pixels* que o GEE utiliza para gerar os histogramas. Esse valor pode ser ajustado em ***“maxPixels”*** e, caso não seja, o limite utilizado é de 10⁶.

Prosseguindo na criação do gráfico, alguns ajustes podem ser feitos, como o uso do método ***“setSeriesNames()”***, do objeto que representa o gráfico (do tipo ***“ui.Chart”***). Ele é usado para retornar um novo gráfico idêntico ao objeto, mas com os nomes das variáveis com os dados (no caso cada banda) definidos na lista de strings passada como parâmetro. Finalmente o método ***“setOptions()”*** é utilizado para passar opções de formatação e apresentação para o gráfico. No exemplo, apenas o título é ajustado com o parâmetro ***“title”***³.

Figura 10 - Criação de um histograma para apenas uma banda

² Mais informações sobre o conceito de escala, no GEE, no link:

<https://developers.google.com/earth-engine/guides/scale>

³ Mais informações sobre estilos, nos gráficos do GEE, podem ser encontradas no link:

https://developers.google.com/chart/interactive/docs/customizing_charts

```
62 //Gera histograma com a distribuição da banda B2.  
63 print(  
64     ui.Chart.image.histogram({  
65         image: imgHistogramaNaoEscalonada.select('B2'),  
66         //maxBuckets: 16,  
67         minBucketWidth: 0.1  
68     })  
69     .setSeriesNames([''])  
70     .setOptions({  
71         title: 'Histograma para reflectância em B2',  
72     })  
73 );
```

```
//Gera histograma com a distribuição da banda B2.  
print(  
    ui.Chart.image.histogram({  
        image: imgHistogramaNaoEscalonada.select('B2'),  
        //maxBuckets: 16,  
        minBucketWidth: 0.1  
    })  
    .setSeriesNames([''])  
    .setOptions({  
        title: 'Histograma para reflectância em B2',  
    })  
);
```

Como exemplo adicional, na Figura 11, é apresentado um código (que também pode ser copiado e adaptado diretamente no *script* criado) para gerar um histograma com todas as bandas simultaneamente. Nesse caso, o intuito é comparar as distribuições dos valores dos *pixels*. O parâmetro “*minBucketWidth*” é ajustado para um valor menor, fazendo com que mais barras sejam geradas e uma aproximação da distribuição de densidade dos *pixels* seja melhor observada. Compare os gráficos resultantes com os valores de estatísticas de cada banda, gerados anteriormente. Note o detalhe do uso do símbolo de adição (+) para concatenar uma string dividida em duas linhas, prática comum para melhorar a legibilidade de um código.

Figura 11 - Criação de histogramas para todas as bandas

```
75 // Gera histograma com as distribuições de todas as bandas.
76 print(
77     ui.Chart.image.histogram({
78         image:imgHistogramaNaoEscalonada,
79         minBucketWidth: 0.01
80     })
81     .setSeriesNames(['B2', 'B3', 'B4', 'B8'])
82     .setOptions({
83         title: 'Histograma para a reflectância' +
84             'em todas as bandas',
85         hAxis: {'title': 'Reflectância'},
86         vAxis: {'title': 'Frequência'},
87     })
88 );
```

```
// Gera histograma com as distribuições de todas as bandas.
print(
    ui.Chart.image.histogram({
        image:imgHistogramaNaoEscalonada,
        minBucketWidth: 0.01
    })
    .setSeriesNames(['B2', 'B3', 'B4', 'B8'])
    .setOptions({
        title: 'Histograma para a reflectância' +
            'em todas as bandas',
        hAxis: {'title': 'Reflectância'},
        vAxis: {'title': 'Frequência'},
    })
);
```

3.6 Ajustando o contraste de uma imagem no GEE

O GEE efetua o ajuste linear baseado em histograma (*stretch*) de forma automática, bastando ajustar o mínimo e máximo. Esses valores podem tanto ser inseridos via código, quanto alterados pela interface através da manipulação das propriedades de um *layer* com uma imagem já adicionada.

Para ajustar de forma automática utilizando a distribuição dos dados dos *pixels*, pode-se expandir o menu **“Layers”** no mapa passando o cursor do mouse em cima do mesmo e em seguida clicando na engrenagem ao lado do nome do *layer* que se deseja alterar - Figura 12. Na nova janela, basta listar as opções no menu *dropdown* do setor **“Range”** - Figura 13. Cada opção vai ajustar o mínimo e máximo baseados na distribuição dos valores dos *pixels*, algumas baseadas no

desvio padrão e outras nos percentis inferior e superior. A Figura 14 apresenta um exemplo de imagem no mapa com o contraste ajustado.

Figura 12 - Listagem das camadas adicionadas no mapa

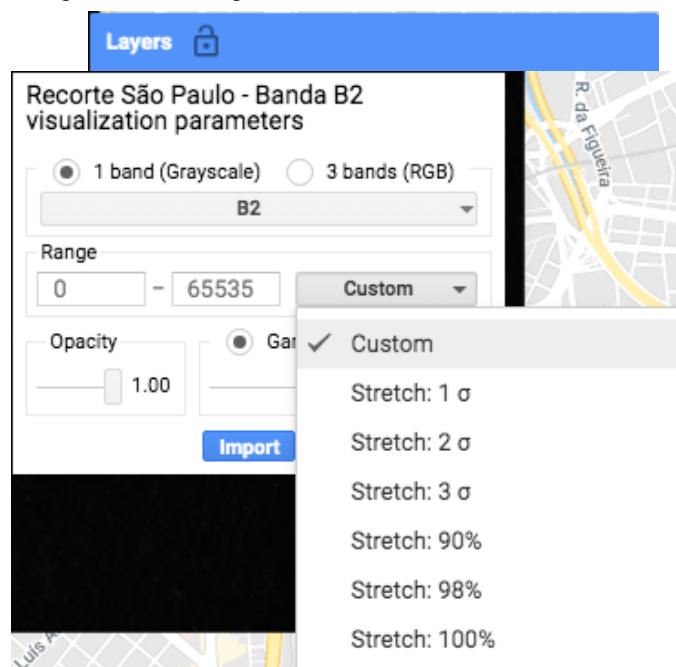
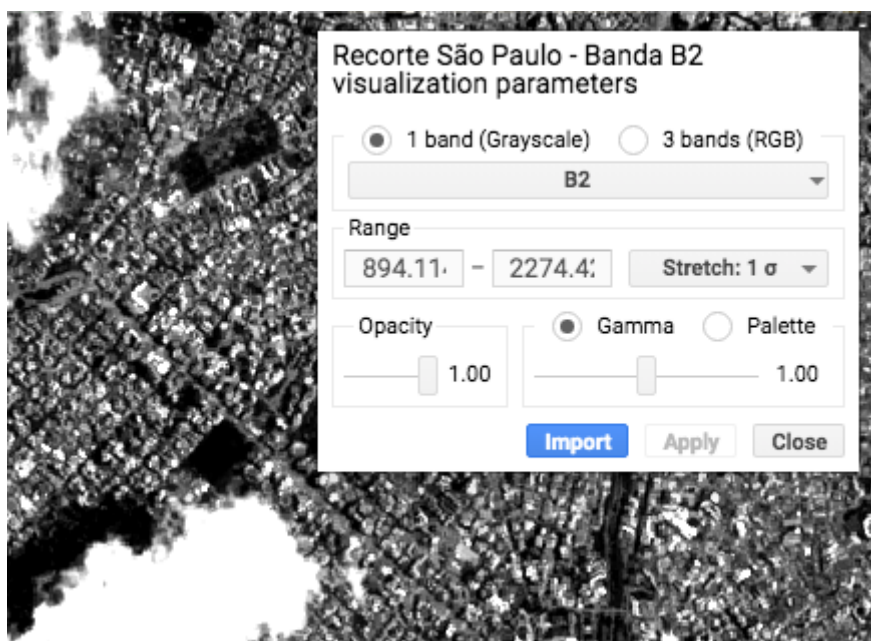


Figura 13 - Opções para se fazer o ajuste linear de contraste no GEE automaticamente

Figura 14 - Imagem com o contraste ajustado



Para fazer o mesmo tipo de ajuste através de código, basta utilizar a função ***“Map.addLayer()”*** com os parâmetros de visualização ***“min”*** (mínimo) e ***“max”***

(máximo) ajustados para os valores desejados. Na próxima seção, alguns parâmetros de visualização de **“Map.addLayer()”** serão melhor compreendidos.

3.7 Composição colorida com ajuste de contraste simples no GEE

Para montar uma composição de bandas e adicionar a imagem ao GEE, deve-se informar à função **“Map.addLayer()”** as bandas que serão utilizadas, caso contrário as três primeiras bandas da imagem serão automaticamente selecionadas e o GEE deduzirá o que fazer com elas. No exemplo da Figura 15 é feita uma composição falsa-cor. O primeiro parâmetro é a imagem de onde se deseja extrair os dados. Já o segundo parâmetro é um dicionário de dados que contém os parâmetros de visualização. Na chave **“bands”** são passados, em uma *string*, os nomes das bandas separados por vírgula. Esses nomes são mapeados na ordem para os canais *RGB* (vermelho, verde e azul) que serão exibidos no mapa. Na sequência os parâmetros **“min”** e **“max”** são mapeados, respectivamente, para os níveis de intensidade 0 e 255. Esses parâmetros são fornecidos para resultar em uma imagem ao lado do cliente do serviço, que é a exibição no navegador. Esse tipo de processamento difere do usual do GEE, que ocorre do lado do servidor. Para escolher os valores de **“min”** e **“max”**, uma alternativa é utilizar os histogramas gerados anteriormente e escolher visualmente os pontos de corte. Nesse caso, um único valor de mínimo e um único de máximo são utilizados para todas as bandas, porém é possível mapear diferentes mínimos e máximos para cada banda.

Figura 15 - Adicionando uma imagem falsa-cor no GEE

```
93 Map.addLayer(  
94     imgHistogramaNaoEscalonada,  
95     {bands:"B8,B4,B3", min:0.04, max:0.4},  
96     "Recorte São Paulo - Composição falsa-cor no recorte"  
97 );
```

```
Map.addLayer(  
    imgHistogramaNaoEscalonada,  
    {bands:"B8,B4,B3", min:0.04, max:0.4},  
    "Recorte São Paulo - Composição falsa-cor no recorte"  
);
```

3.8 Indo além

3.8.1 Composição colorida com ajustes de contraste individuais no GEE

Como mencionado no parágrafo anterior, é possível fazer um mapeamento mais preciso de máximos e mínimos para cada banda, e assim ajustar de forma mais apropriada o contraste da imagem que será exibida na interface do GEE. Além disso, para fazer o procedimento de forma mais acertada, ao invés de se fazer uma escolha visual, pode-se utilizar as estatísticas das distribuições de cada banda e escolher os máximos e mínimos de acordo com medidas como médias e desvios padrão das bandas. O modo de se calcular essas estatísticas já foram apresentados anteriormente. Para se calcular o valor máximo, com um desvio padrão além da média, pode-se executar um código similar ao apresentado na Figura 16. Aqui, é apresentado um padrão mais avançado de codificação do GEE, com uma função aninhada dentro da outra. A função mais externa, ***“calculaMaxDp()”*** recebe como parâmetro o número de desvios padrão que serão usados para calcular o máximo (***“totalDp”***). Esse parâmetro é passado junto da função na chamada do método ***“map()”*** do objeto que contém as médias. Como esse objeto é um dicionário, o ***“map()”*** carrega a chave e valor de cada elemento do dicionário. Essas duas variáveis serão passadas para a função mais interna (***“calcula()”***) quando a linha com o ***“return calcula;”*** for executada. Nesse caso, a chamada trará a função declarada logo acima na variável ***“calcula”***. Dentro da função ***“calcula()”*** o cálculo é propriamente efetuado. Deve ser efetuada uma transformação no tipo de dado do valor para ***“ee.Number”***, antes de se calcular a adição com o ***“add()”***. O valor a ser adicionado é retirado do dicionário de dados que contém os desvios padrão das bandas. Com o método ***“getNumber()”*** aplicado na chave (que é a mesma do dicionário de dados das médias) é feita a extração do valor e finalmente o número de desvios padrão desejados no deslocamento é calculado com o método ***“multiply()”***. De forma similar, podem ser calculados os valores mínimos para se mapear as bandas, com o método ***“subtract()”*** (Figura 17).

Figura 16 - Calculando os valores máximos para mapear cada banda


```
// Adiciona o número total de desvios padrões desejados à média.
var calculaMaxDp = function (totalDp) {
    var calcula = function (chave, valor) {
        return ee.Number(valor).add(dpPixelsBandas
                                .getNumber(chave)
                                .multiply(totalDp)
                                );
    };
    return calcula;
};
// Processa a função para calcular o limite superior
// do ajuste de contraste para todas as bandas.
var maxIntervalos = mediaPixelsBandas.map(calculaMaxDp(1));
```

Figura 17 - Calculando os valores mínimos para mapear cada banda

```
// Retira o número total de desvios padrões desejados da média.
var calculaMinDp = function (totalDp) {
    var calcula = function (chave, valor) {
        return ee.Number(valor).subtract(dpPixelsBandas
                                .getNumber(chave)
                                .multiply(totalDp)
                                );
    };
    return calcula;
};
// Processa a função para calcular o limite inferior
// do ajuste de contraste para todas as bandas.
var minIntervalos = mediaPixelsBandas.map(calculaMinDp(1));
```

Para o procedimento que adiciona a imagem no mapa, deve-se pegar os máximos e mínimos calculados e criar um dicionário de dados com todos eles, como apresentado na Figura 18. Na primeira linha, é criada uma lista a partir de duas listas, que são os valores extraídos com o método “**values()**” dos dicionários de dados com os mínimos e máximos. Com “**ee.List()**”, é criado um objeto do tipo lista, com duas listas com os valores de mínimos em uma e de máximos na outra. Finalmente é disparado o método “**flatten()**” que remove os elementos das sub-listas e cria uma lista na sequência em que os dados estavam nas listas iniciais, sendo os primeiros elementos os da primeira lista, seguidos então pelos da segunda. Na segunda linha da figura, é criada uma lista com as *strings* com as

respectivas chaves que serão utilizadas para se referir aos valores, combinando nessa chave os nomes das bandas e se são mínimo ou máximo. Finalmente, na última linha é criado o dicionário a partir das listas com a função

“*ee.Dictionary.fromLists()*”, onde o primeiro parâmetro é a lista com as chaves e o segundo a lista com os valores.

Figura 18 - Criação de dicionário de dados com todos os valores de mínimos e máximos

```
var valoresMinimosMaximos = ee.List([minIntervalos.values(),  
                                     maxIntervalos.values()]).flatten();  
var chavesMinimosMaximos = ['B2_min', 'B3_min', 'B4_min', 'B8_min',  
                           'B2_max', 'B3_max', 'B4_max', 'B8_max'];  
var dictMinimosMaximos = ee.Dictionary.fromLists(chavesMinimosMaximos,  
                                                  valoresMinimosMaximos);
```

Com os dados para mapear os mínimos e máximos prontos, basta efetuar o procedimento da Figura 19 para adicionar a imagem ao mapa. O código apresentado na figura apresenta mais algumas novidades. A chamada para se adicionar a imagem ao mapa está dentro de uma função chamada **“*exibeComposicaoColorida()*”**, que carrega como único parâmetro o objeto que chama a função. Essa chamada ocorre através do método **“*evaluate()*”**. O truque utilizado aqui, com o **“*evaluate()*”**, deve ser feito pelo fato levantado anteriormente de o GEE trabalhar em duas camadas, o lado do servidor e o lado do cliente (*browser*). Variáveis criadas de um lado não interagem com o outro, causando erros. Não seria possível passar os valores calculados de mínimos e máximos diretamente para a função **“*Map.addLayer()*”**, pois ela é executada no cliente e os valores estão em um objeto no servidor. O que o método **“*evaluate()*”** faz, é trazer os dados do objeto que o chama do servidor para o cliente e enviá-los para a função passada como parâmetro - no caso **“*exibeComposicaoColorida()*”**. A função é então executada e consegue utilizar os dados outrora no lado do servidor. No exemplo, dentro da função **“*exibeComposicaoColorida()*”**, é chamada a função **“*Map.addLayer()*”** com o primeiro parâmetro ajustado para a imagem (nesse caso, embora o parâmetro seja um objeto do servidor, o cliente faz os *download* dos dados automaticamente) e o segundo um dicionário de dados com as configurações. Aqui, **“*bands*”** recebe uma lista (valores entre colchetes e separados por vírgula, não um

objeto “**ee.List()**”) com as bandas na sequência de mapeamento para *RGB* (no exemplo anterior foi mostrada a opção de se usar uma *string*). Finalmente, “**min**” e “**max**”, são alimentadas com listas com os valores retirados do dicionário de dados que chamou “**evaluate()**”. A notação utiliza o nome do dicionário dentro da função, seguindo de um ponto e a chave.

Figura 19 - Adicionando a imagem no mapa com parâmetros para cada banda

```
var exibeComposicaoColorida = function(dictMinimosMaximos) {  
  Map.addLayer(imgHistogramaNaoEscalonada,  
    {  
      bands: ["B4", "B3", "B2"],  
      min: [dictMinimosMaximos.B4_min,  
            dictMinimosMaximos.B3_min,  
            dictMinimosMaximos.B2_min],  
      max: [dictMinimosMaximos.B4_max,  
            dictMinimosMaximos.B3_max,  
            dictMinimosMaximos.B2_max]  
    },  
    "Recorte São Paulo - Composição Colorida"  
  );  
};  
dictMinimosMaximos.evaluate(exibeComposicaoColorida);
```

3.8.2 Trabalhando com *DN*

Todos os procedimentos efetuados na atividade podem ser feitos com coleções que trazem os números digitais em seus *pixels*. Fica como exercício extra testar e analisar alguma coleção do tipo *Raw Data* produzida por alguma missão *Landsat*.

4 Conclusões

Neste laboratório foram apresentados, de forma prática no GEE, como trabalhar com estatísticas e explorar os dados dos pixels de uma imagem. Também foram introduzidas formas de se trabalhar com o ajuste de contraste de imagens no GEE, tanto de forma interativa quanto por código.

5 Referências bibliográficas

CRÓSTA, A. P. Processamento Digital de Imagens de Sensoriamento Remoto. Campinas: Ed. Rev. Campinas, UNICAMP. 1992.