

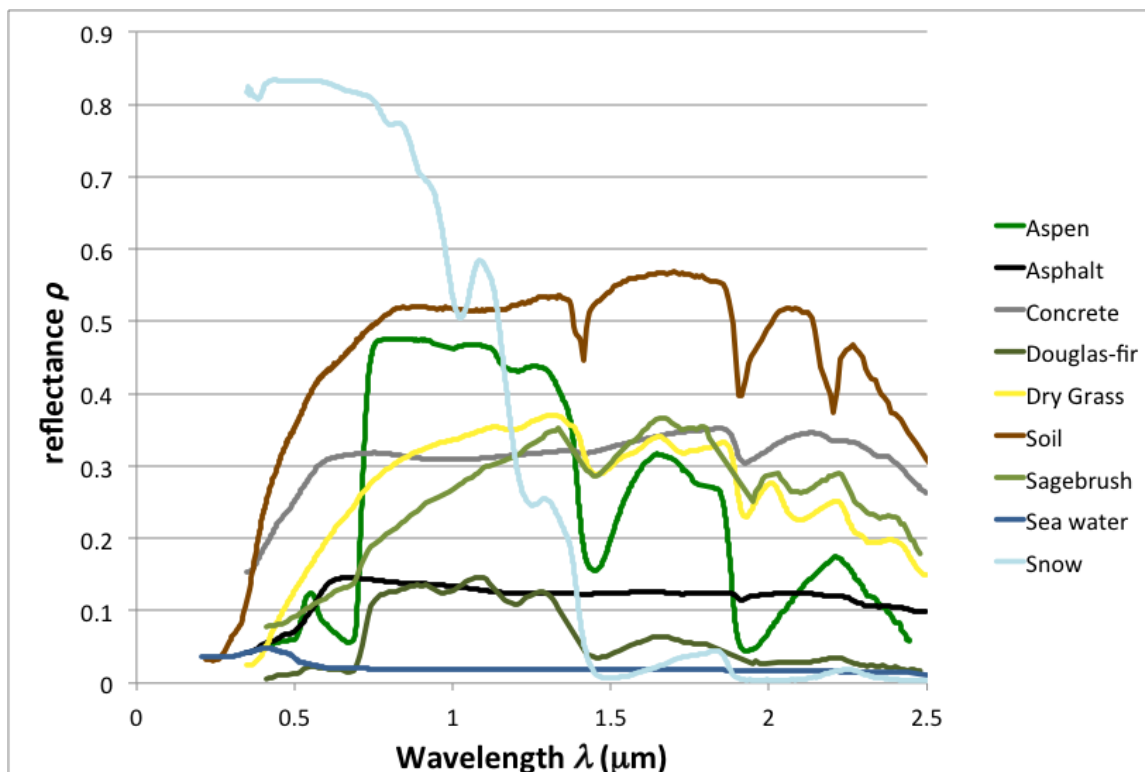
Lab. 4 : Indices spectraux des images satellitaires

Objectifs : Le but de ce laboratoire est de vous donner une introduction de l'implémentation des indices spectraux des images satellitaires. Vous serez initié aux méthodes de création des indices de végétation, d'eau, de neige, d'indices de sol nu et de zones brûlées. À la fin du laboratoire, vous serez en mesure de mettre en œuvre des indices spectraux et des transformations pour accentuer l'information d'intérêt dans votre domaine d'étude.

Prerequisites: Lab 3

1. Indices spectraux

Les indices spectraux sont basés sur le fait que les spectres de réflectance des différentes occupations du sol sont différents. Les indices sont conçus pour exploiter ces différences afin d'accentuer certains types de d'occupation du sol. Considérez la figure suivante des spectres de réflectance pour diverses occupations du sol :



Observez que les occupations du sol sont séparables à une ou plusieurs longueurs d'onde. Il convient de noter, en particulier, que les courbes de végétation (vert) ont une réflectance relativement élevée dans la gamme NIR, où l'énergie radiante est dispersée par les parois cellulaires ([Bowker et al., 1985](#)). Notez également que la végétation a une faible réflectance dans la gamme rouge, où l'énergie radiante est [absorbée par la chlorophylle](#). Ces observations motivent la formulation d'indices de végétation, par exemple :

a. NDVI.

L'indice normalisé de végétation de différence (NDVI) a une [longue histoire](#) de télédétection. La formulation typique est

$$NDVI = (NIR - red) / (NIR + red)$$

Lorsque *le NIR* et le *rouge* se réfèrent à la réflectance, l'éclat ou le DN à la longueur d'onde respective. Implémenter des indices dans Earth Engine avec la fonction ou méthode `normalizedDifference()`. Tout d'abord, obtenez une image d'intérêt en dessinant un point nommé, en important la Collection Landsat 8 1 TOA comme `landsat8` et en triant la collection par métadonnées de couverture nuageuse :

```
var image = ee.Image(landsat8
  .filterBounds(point)
  .filterDate('2015-06-01', '2015-09-01')
  .sort('CLOUD_COVER')
  .first());
```

```
var trueColor = {bands: ['B4', 'B3', 'B2'], min: 0, max: 0.3};
Map.addLayer(image, trueColor, 'image');
```

Le calcul NDVI est une ligne :

```
var ndvi = image.normalizedDifference(['B5', 'B4']);
```

Affichez l'image NDVI avec une palette de couleurs (n'hésitez pas à en faire une meilleure) :

```
var vegPalette = ['white', 'green'];
Map.addLayer(ndvi, {min: -1, max: 1, palette: vegPalette}, 'NDVI');
```

Utilisez l'**inspecteur** pour vérifier les valeurs des pixels dans les zones de végétation et de non-végétation.

b. EVI.

L'indice de végétation amélioré (EVI) est conçu pour minimiser les effets de saturation et de fond dans NDVI ([Huete et al., 2002](#)). Comme il ne s'agit pas d'un indice de différence normalisé, calculez-le avec [une expression](#):

```
var evi = image.expression(  
  '2.5 * ((NIR - RED) / (NIR + 6 * RED - 7.5 * BLUE + 1))', {  
    'NIR': image.select('B5'),  
    'RED': image.select('B4'),  
    'BLUE': image.select('B2')  
  });
```

Observez que les bandes sont référencées à l'aide [d'un objet](#) qui est passé comme le deuxième argument `image.expression()`. Affichage de EVI:

```
Map.addLayer(evi, {min: -1, max: 1, palette: vegPalette}, 'EVI');
```

Comparez EVI à NDVI. Qu'observez-vous ?

c. NDWI.

L'indice normalisé de l'eau de différence (NDWI) a été élaboré par [Gao \(1996\)](#) comme un indice de la teneur en eau de végétation :

$$NDWI = (NIR - SWIR) / (NIR + SWIR)$$

Calculer NDWI dans Earth Engine avec :

```
var ndwi = image.normalizedDifference(['B5', 'B3']);
```

Et afficher :

```
var waterPalette = ['white', 'blue'];  
Map.addLayer(ndwi, {min: -0.5, max: 1, palette: waterPalette},  
  'NDWI');
```

Notez qu'il ne s'agit pas d'une implémentation exacte de NDWI, selon la [réponse spectrale OLI](#), puisque OLI n'a pas de bande dans la bonne position (1,26 µm).

d. NDWBI.

Il est regrettable *different* que deux indices NDWI différents aient été inventés indépendamment en 1996. Pour distinguer, définissez l'indice normal de la différence de l'eau dans l'eau (NDWBI) comme l'indice décrit dans [McFeeters \(1996\)](#):

$$\text{NDWBI} = (\text{green} - \text{NIR}) / (\text{green} + \text{NIR})$$

Comme précédemment, implémentez NDWBI avec la méthode `normalizedDifference()` et affichez le résultat :

```
var ndwi = image.normalizedDifference(['B3', 'B5']);  
Map.addLayer(ndwi, {min: -1, max: 0.5, palette: waterPalette},  
'NDWBI');
```

Comparez NDWI et NDWBI. Qu'observez-vous ?

e. NDBI.

L'indice normalisé de la différence nue (NDBI) a été élaboré par [Zha et coll. \(2003\)](#) pour faciliter la différenciation des zones urbaines :

$$\text{NDBI} = (\text{SWIR} - \text{NIR}) / (\text{SWIR} + \text{NIR})$$

Notez que NDBI est le négatif de NDWI. Calculer NDBI et afficher avec une palette appropriée :

```
var ndbi = image.normalizedDifference(['B6', 'B5']);  
var barePalette = waterPalette.slice().reverse();  
Map.addLayer(ndbi, {min: -1, max: 0.5, palette: barePalette}, 'NDBI');
```

(Vérifiez [cette référence](#) pour démystifier l'inversion de la palette).

f. BAI.

L'indice de la zone brûlée (BAI) a été élaboré par [Chuvieco et coll. \(2002\)](#) pour aider à la délimitation des cicatrices de brûlures et à l'évaluation de la gravité des brûlures. Il est basé sur la distance spectrale à la réflectance charbon de bois. Pour examiner les indices de brûlure, chargez une image de 2013 montrant l'incendie [de Rim](#) dans la Sierra Nevada :

```
var burnImage = ee.Image(landsat8
    .filterBounds(ee.Geometry.Point(-120.083, 37.850))
    .filterDate('2013-08-17', '2013-09-27')
    .sort('CLOUD_COVER')
    .first());
```

```
Map.addLayer(burnImage, trueColor, 'burn image');
```

Examinez de près l'affichage de couleur réel de cette image. Pouvez-vous repérer le feu ? Si ce n'est pas le cas, le BAI peut vous aider. Comme avec EVI, utilisez une expression pour calculer BAI dans Earth Engine :

```
var bai = burnImage.expression(
    '1.0 / ((0.1 - RED)**2 + (0.06 - NIR)**2)', {
    'NIR': burnImage.select('B5'),
    'RED': burnImage.select('B4'),
});
```

Affichez le résultat. La zone de brûlure devrait être plus évidente dans la visualisation BAI.

```
var burnPalette = ['green', 'blue', 'yellow', 'red'];
Map.addLayer(bai, {min: 0, max: 400, palette: burnPalette}, 'BAI');
```

g. NBRT.

Le ratio de combustion normalisé thermique (NBRT) a été développé sur la base de l'idée que les terres brûlées ont une faible réflectance NIR (moins de végétation), haute réflexion SWIR (pensez cendres), et la température de luminosité élevée ([Holden et al. 2005](#)). Contrairement aux autres indices, une NBRT plus faible signifie plus de combustion. Implémenter le NBRT avec une expression :

```
var nbrt = burnImage.expression(
    '(NIR - 0.0001 * SWIR * Temp) / (NIR + 0.0001 * SWIR * Temp)', {
```

```
'NIR': burnImage.select('B5'),
'SWIR': burnImage.select('B7'),
'Temp': burnImage.select('B11')
});
```

Pour afficher ce résultat, inversez l'échelle :

```
Map.addLayer(nbrt, {min: 1, max: 0.9, palette: burnPalette}, 'NBRT');
```

La différence dans cet indice, avant - après l'incendie, peut être utilisée comme diagnostic de la gravité des brûlures (voir [van Wagtendonk et al. 2004](#)).

h. **NDSI.**

L'indice normalisé de la différence de neige (NDSI) a été conçu pour estimer la quantité d'un pixel couvert de neige ([Riggs et al. 1994](#)) :

$$\text{NDSI} = (\text{green} - \text{SWIR}) / (\text{green} + \text{SWIR})$$

Tout d'abord, trouver une scène couverte de neige pour tester l'index:

```
var snowImage = ee.Image(landsat8
  .filterBounds(ee.Geometry.Point(-120.0421, 39.1002))
  .filterDate('2013-11-01', '2014-05-01')
  .sort('CLOUD_COVER')
  .first());
```

```
Map.addLayer(snowImage, trueColor, 'snow image');
```

Calculer et afficher NDSI dans Earth Engine :

```
var ndsi = snowImage.normalizedDifference(['B3', 'B6']);
var snowPalette = ['red', 'green', 'blue', 'white'];
Map.addLayer(ndsi, {min: -0.5, max: 0.5, palette: snowPalette},
  'NDSI');
```

a. Spectral unmixing. (facultatif)

Le [modèle de mélange spectral linéaire](#) est basé sur l'hypothèse que chaque pixel est un mélange de spectres « pur ». Les spectres purs, appelés endmembers, proviennent de classes de couverture terrestre telles que l'eau, la terre nue, la végétation. L'objectif est de résoudre l'équation suivante pour f , le vecteur $P \times 1$ des fractions endmembers dans le pixel :

$$Sf = p$$

où S est une matrice $B \times P$ dans laquelle les colonnes sont des spectres de membre d'end pur p (connu) et p est le vecteur de pixels $B \times 1$ lorsqu'il y a des bandes $B \times P$ (connues). Dans cet exemple, $B=6$:

```
var unmixImage = image.select(['B2', 'B3', 'B4', 'B5', 'B6', 'B7']);
```

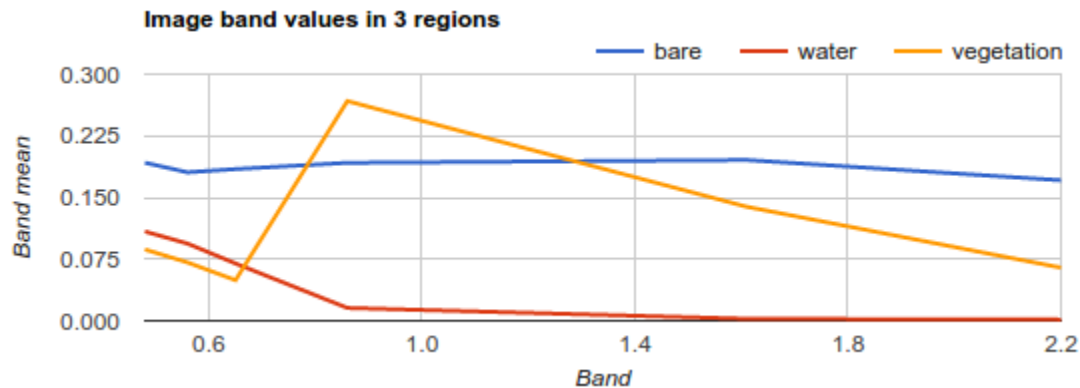
La première étape consiste à obtenir les spectres du membre final. Faites-le en calculant les spectres moyens dans les polygones délimités autour des régions de couverture terrestre pure. Zoomez sur la carte jusqu'à un endroit où les zones homogènes de terre nue, de végétation et d'eau (indice : SFO). Visualisez l'entrée sous la forme d'un faux composite de couleur :

```
Map.addLayer(image, {bands: ['B5', 'B4', 'B3'], max: 0.4}, 'false color');
```

À l'aide des [outils de dessin de géométrie](#), faire trois nouvelles couches ($P=3$) en cliquant + nouveau **calque (+new layer)**. Dans la première couche, numérisez un polygone autour d'une terre nue pure; dans la deuxième couche faire un polygone de végétation pure; dans la troisième couche, faire un polygone de l'eau. Nommer les importations **bare, veg, and water**, Respectivement. Vérifiez les polygones que vous avez faits en traçant les spectres moyens en les utilisant [Chart.image.regions\(\)](#):

```
print(Chart.image.regions(unmixImage, ee.FeatureCollection([
  ee.Feature(bare, {label: 'bare'}),
  ee.Feature(water, {label: 'water'}),
  ee.Feature(veg, {label: 'vegetation'})])),
  ee.Reducer.mean(), 30, 'label', [0.48, 0.56, 0.65, 0.86, 1.61,
2.2]));
```

Votre graphique devrait ressembler à :



Utiliser le [reduceRegion\(\) method](#) pour calculer les spectres moyens dans les polygones que vous avez fait. Notez que la valeur de retour de [reduceRegion\(\)](#) est un [Dictionary](#), avec sortie réducteur clé par nom de bande. Obtenez les moyens en tant que liste en appelant [values\(\)](#):

```
var bareMean = unmixImage
    .reduceRegion(ee.Reducer.mean(), bare, 30).values();
var waterMean = unmixImage
    .reduceRegion(ee.Reducer.mean(), water, 30).values();
var vegMean = unmixImage
    .reduceRegion(ee.Reducer.mean(), veg, 30).values();
```

Chacune de ces trois listes représente un vecteur de spectre moyen. Empilez les vecteurs dans un tableau 6x3 de membres d'extrémité en les concaténant le long de l'axe 1 (ou direction des colonnes) :

```
var endmembers = ee.Array.cat([bareMean, vegMean, waterMean], 1);
```

Transformez l'image d'entrée en 6 bandes en une image dans laquelle chaque pixel est un vecteur 1D ([toArray\(\)](#)), puis dans une image dans laquelle chaque pixel est une matrice 6x1 ([toArray\(1\)](#)):

```
var arrayImage = unmixImage.toArray().toArray(1);
```

Maintenant que les dimensions correspondent, dans chaque pixel, résoudre l'équation pour **f**:

```
var unmixed = ee.Image(endmembers).matrixSolve(arrayImage);
```


Enfin, convertir le résultat d'une image de tableau 2D en une image de tableau 1D (`arrayProject()`), puis à une image multi-bande (`arrayFlatten()`). Les trois bandes correspondent aux estimations des fractions de sol nu, de la végétation et de l'eau `f`:

```
var unmixedImage = unmixed
    .arrayProject([0])
    .arrayFlatten(['bare', 'veg', 'water']));
```

Afficher le résultat où le sol nu est rouge, la végétation est verte, et l'eau est bleue (the `addLayer()` appel attend bandes dans l'ordre, RGB):

```
Map.addLayer(unmixedImage, {}, 'Unmixed');
```

1. The HSV transform

Le modèle Hue-Saturation-Value (HSV) [est une transformation de couleur de l'espace de couleur RGB](#). Entre autres choses, il est utile pour le [pan-affûtage](#). Il s'agit de convertir un RGB en HSV, swapping the panchromatic band for the value (V), then converting back to RGB. For example, using the Landsat 8 scene:

```
// Convertir les bandes Landsat RGB en HSV
var hsv = image.select(['B4', 'B3', 'B2']).rgbToHsv();
Convertir en RGB, en échangeant la bande panchromatique d'image pour la valeur.
```

```
var rgb = ee.Image.cat([
  hsv.select('hue'),
  hsv.select('saturation'),
  image.select(['B8'])
]).hsvToRgb();
```

```
Map.addLayer(rgb, {max: 0.4}, 'Pan-sharpened');
```

Comparez l'image pan-aiguisée à l'image d'origine.



This work is licensed under a [Creative Commons Attribution 4.0 International License](#).
Copyright 2015-2016, Google Earth Engine Team