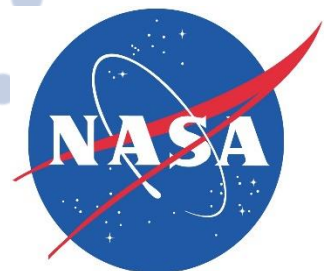


Tutorial #1  
Plotting the Ratio of TROPOMI  
Formaldehyde and Nitrogen Dioxide  
Satellite Data: An Indicator of Ozone  
Sensitivity

Jennifer McGinnis  
Tracey Holloway  
Jenny Bratburd

*Nelson Institute Center for sustainability and the Global Environment  
University of Wisconsin-Madison  
Madison, WI 53726*



## • • • • • **Acknowledgements**

Work for this report was funded by the NASA Health and Air Quality Applied Sciences Team (HAQAST).

We thank all the groups that make their data publicly available for air quality and health analysis, especially those we used in this report: NASA TROPOMI and Earth Engine Data Catalog. In addition, we would like to thank the broader NASA HAQAST team for helpful comments at various stages of the project.

# CONTENTS

<b>Acknowledgements</b> .....	2
<b>Executive Summary</b> .....	4
<b>Introduction</b> .....	4
<b>Google Earth Engine and TROPOMI Background</b> .....	5
<b>Tutorial and Case Study</b> .....	5
<b>Conclusion</b> .....	14
<b>Code</b> .....	15
<b>References</b> .....	16

## • • • • • Executive Summary

Satellite-derived data can complement state agencies' ground monitor-based approach for improving air quality. This method allows for continuous data coverage at high resolutions to provide a broader coverage of air pollutants. This document will provide guidance for using formaldehyde (HCHO) and nitrogen dioxide (NO<sub>2</sub>) satellite data to help visualize ground-level ozone sensitivity. An introduction to this method along with the code used in a case study over Wisconsin is below. All that is needed for this tutorial is a computer with access to the internet and a Google account.

Key Aspects:

- Setting up Google Earth Engine
- Identifying the region of interest and datasets
- Calculating the average of HCHO and NO<sub>2</sub>
- Plotting the FNR

## • • • • • Introduction

In the stratosphere, ozone is a naturally occurring molecule that protects Earth from the sun's harsh UV rays. However, ozone can also be present in the troposphere, or at ground level, where it is harmful to human health. Due to its health implications, ozone is a criteria air pollutant and is regulated by the National Ambient Air Quality Standards (NAAQS) imposed by the Environmental Protection Agency (EPA).

Most commonly, ozone is formed by volatile organic carbons (VOCs) and nitrogen oxides (NO<sub>x</sub> = NO<sub>2</sub> + NO) reacting in the presence of sunlight. To effectively manage the production of ozone, air quality agencies require an assessment of ozone's sensitivity to emitted VOCs and NO<sub>x</sub> to better understand which pollutant is the limiting component in the formation of ozone. However, this analysis is severely constrained by the absence of ground-based measurement data. Utilizing satellite data provides continuous spatial coverage of NO<sub>2</sub> and formaldehyde (HCHO), a VOC that indicates ozone production. This enables air quality managers to evaluate ozone sensitivity even in areas without monitors (Witman et al., 2014). The ratio of HCHO/NO<sub>2</sub>, can be used to better understand ground level ozone sensitivity and aid in determining mitigation strategies. This ratio is called the Formaldehyde to Nitrogen dioxide Ratio (FNR).

$$\text{FNR} = \frac{\text{HCHO}}{\text{NO}_2}$$

A high FNR is caused by a high HCHO value relative to NO<sub>2</sub>, indicating a NO<sub>x</sub>-limited region (NO<sub>x</sub> are the limiting component in the formation of ozone). A low FNR is caused by a low HCHO value relative to NO<sub>2</sub>, indicating a VOC-limited region (VOCs are the limiting component in the formation of ozone). Additionally, there is a transitional range where a decrease in both NO<sub>x</sub> and VOCs is recommended to decrease ozone formation.

## Google Earth Engine and TROPOMI Background

Google Earth Engine is a computing platform that is currently free of charge for noncommercial and research use. Additionally, the code editor allows for user friendly analysis using the programming language JavaScript. Google Earth Engine has resources available with example code and a catalog of ready to use geospatial datasets and satellite imagery. Some of the available datasets include data products from the Sentinel sensors, MODIS sensors, and Landsat sensors. In this document, we will focus on the Sentinel-5 Precursor (S5P) mission [datasets](#).

The TROPospheric Monitoring Instrument (TROPOMI) aboard the S5P satellite offers daily coverage of the world from July 2018-present. Most satellite instruments take measurements of the entire column of atmosphere from the ground to the instrument. For some pollutants, TROPOMI can separate the tropospheric data from stratospheric data. In this document we will use the tropospheric HCHO and NO<sub>2</sub> retrievals to calculate the FNR. When using TROPOMI satellite data to calculate the FNR, the transitional range we will use is [3.2-4.1] meaning any region with an FNR value less than 3.2 is VOC-limiting and any region with a FNR value more than 4.1 is NO<sub>x</sub>-limiting (Jin et al., 2020). This transitional range was calculated using empirical observation-based analysis over seven cities.

The NO<sub>2</sub> and HCHO data products have specific quality parameters, defined as the "qa\_value", outlined in their user manuals to ensure only high-quality data is plotted. When using the TROPOMI NO<sub>2</sub> data product, it is recommended to ignore data with a qa\_value <0.75 (Eskes et al., 2019). When using the TROPOMI HCHO data product, it is recommended to ignore data with a qa\_value <0.50 (Romahn et al., 2022). Conveniently, Google Earth Engine automatically filters the data to remove pixels with lower values than the qa\_value for both datasets.

One challenge when analyzing HCHO satellite retrievals is that the daily observations are relatively noisy. To account for this weak signal of HCHO, it is recommended to average either temporally or spatially when using TROPOMI HCHO satellite data. In this document, we will average over a few months. After taking the temporal average of each individual pollutant, we will divide the HCHO average by the NO<sub>2</sub> average to determine the FNR. The FNR will then be plotted onto a map.

## Tutorial and Case Study

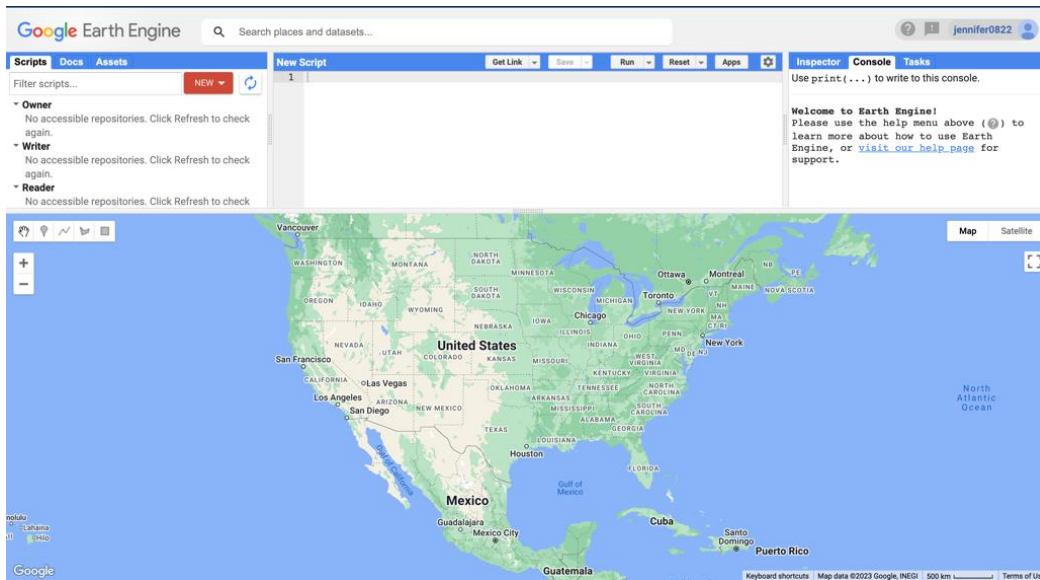
*Last updated and tested on 8/1/2023*

### 1) Set up a Google Earth Engine Account.

- a. Go to [Google Earth Engine](#) and click the "Getting Started" button in the top right corner.
- b. Select or enter your email address.
- c. If required, select "Use with a Cloud Project" → select "Unpaid Usage" → select project type and click the "Next" button.
- d. Next, select "Create a new Google Cloud Project", fill in the organization and project ID, and click "Continue to Summary" → select "Confirm and Continue".

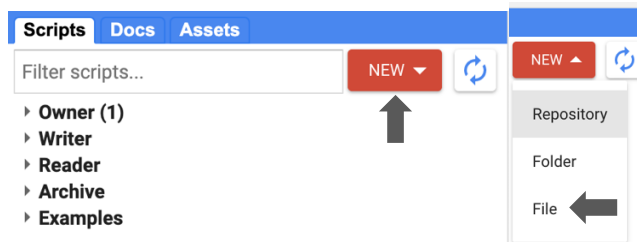
### 2) Navigate to the Google Earth Engine code editor (pictured below).

- a. After creating an account, you may have already been brought to this page. If not, access the [Google Earth Engine](#) website → hover over the "Platform" tab → click "Code Editor". This is where we will load in and analyze our satellite data.



### 3) Create a new file for the script.

- a. Click the red “New” button located near the top left of the screen, then click “File” from the drop-down menu that appeared.



- b. Enter a username and repository name of your choosing.
- c. Enter a file name, such as “FNR”.
- d. Double click on the name of the script to open it

### 4) For convenience, the complete script from this point forward is provided under the “Code” section of this document.

### 5) Define the geographic region of interest using the United States Census Bureau Tiger dataset containing the United States state boundaries.

- a. Note that this step is not required. However, not defining geographic bounds will cause a longer computing time. Additionally, there are many ways to define geographic bounds, as shown [here](#). In this tutorial, the geographic bounds will be defined as the state of Wisconsin.
- b. Create a variable in the code editor named geometry to store the geographic bounds.
  - i. Note: Variables can be named whatever makes sense for the user.

```
FNR
1 var geometry =
```

- c. Go to the [Earth Data Catalog](#) to find the TIGER: US Census States 2018 [dataset](#).
  - i. Locate the Earth Engine Snippet and copy the code provided.

## Earth Engine Snippet

FeatureCollection

```
ee.FeatureCollection("TIGER/2018/States")
```

- ii. Paste the code snippet into the geometry variable.

```
FNR Get Link  
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
```

- d. Define the state(s) of interest.
  - i. Use `.filter(ee.Filter.eq('NAME', '*Insert State Name*))` to specify the state.
    1. For more than one state use:

```
.filter(ee.Filter.or(  
  ee.Filter.eq('NAME', '*Insert First State Name*'),  
  ee.Filter.eq('NAME', '*Insert Second State Name*'))
```
  - ii. Note: a semicolon is added to indicate the end of a variable.

```
FNR Get Link  
1 var geometry = ee.FeatureCollection("TIGER/2018/States")  
2 .filter(ee.Filter.eq('NAME', 'Wisconsin'));
```

### 6) Create a variable named “formaldehyde” to store the formaldehyde satellite data.

```
FNR Get Link Save Run  
1 var geometry = ee.FeatureCollection("TIGER/2018/States")  
2 .filter(ee.Filter.eq('NAME', 'Wisconsin'));  
3  
4 var formaldehyde =
```

- a. In this variable, three things will be specified: the dataset of interest, the band of interest from the dataset, and the time frame of interest.
- b. Define the dataset.
  - i. Go to the [Earth Data Catalog](#) to find the TROPOMI Formaldehyde [dataset](#). There is an “Offline” dataset indicated with “OFFL” in the dataset name or there is a “Near Real Time” dataset indicated with “NRTI” in the dataset name. We will use the offline dataset.
  - ii. Locate the Earth Engine Snippet and copy the code provided.

#### Earth Engine Snippet

```
ee.ImageCollection("COPERNICUS/S5P/OFFL/L3_HCHO")
```

- iii. Then paste the code snippet into the formaldehyde variable.

```
FNR Get Link Save Run Reset  
1 var geometry = ee.FeatureCollection("TIGER/2018/States")  
2 .filter(ee.Filter.eq('NAME', 'Wisconsin'));  
3  
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
```

- c. Define the band.
  - i. In the TROPOMI Formaldehyde dataset, click on the “[Bands](#)” tab.

- ii. Only the Tropospheric HCHO column number density is needed. Copy the name of the band that is provided in the Bands tab.
- iii. Use `.select(*insert band name*)` to define the band of interest from the TROPOMI Formaldehyde dataset in the formaldehyde variable.

```

FNR Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')

```

- d. Define the time frame.
  - i. In this tutorial, the time frame of interest are the months April, May, June, July, August, and September of 2019.
    1. Start date: 2019-04-01 (Inclusive)
    2. End date: 2019-10-01 (Exclusive – meaning the last day used in the analysis is September 30<sup>th</sup>)
  - ii. Use `.filterDate(*Start Date*, *End Date*)` to define the time frame within the formaldehyde variable. End the variable with a semicolon.

```

FNR Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')
6   .filterDate('2019-04-01', '2019-10-01');

```

7) Create a variable named “no2” to store the nitrogen dioxide satellite data.

```

FNR Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')
6   .filterDate('2019-04-01', '2019-10-01');
7
8 var no2 =

```

- a. In this variable, three things will be specified: the dataset of interest, the band of interest from the dataset, and the time frame of interest.
- b. Define the dataset.
  - i. Go to the [Earth Data Catalog](#) to find the TROPOMI NO<sub>2</sub> [dataset](#).
  - ii. Locate the Earth Engine Snippet and copy the code provided.

**Earth Engine Snippet**

```
ee.ImageCollection("COPERNICUS/S5P/OFFL/L3_N02")
```



iii. Paste the code snippet into the no2 variable.

```
FNR Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')
6   .filterDate('2019-04-01', '2019-10-01');
7
8 var no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
```

c. Define the band.

- i. In the TROPOMI NO<sub>2</sub> dataset, click on the “[Bands](#)” tab.
- ii. Only the Tropospheric NO<sub>2</sub> column number density is needed so copy the name of the band that is provided in the Bands tab.
- iii. Use `.select(*insert band name*)` to define the band of interest from the TROPOMI Formaldehyde dataset in the formaldehyde variable.

```
FNR Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')
6   .filterDate('2019-04-01', '2019-10-01');
7
8 var no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
9   .select('tropospheric_NO2_column_number_density')
```

d. Define the time frame.

- i. This will be the same as the time frame for formaldehyde: April, May, June, July, August, and September of 2019.
  1. Start date: 2019-04-01 (Inclusive)
  2. End date: 2019-10-01 (Exclusive – meaning the last day used in the analysis is September 30<sup>th</sup>)
- ii. Use `.filterDate(*Start Date*, *End Date*)` to define the time frame within the formaldehyde variable. End the variable with a semicolon.

```
FNR Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')
6   .filterDate('2019-04-01', '2019-10-01');
7
8 var no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
9   .select('tropospheric_NO2_column_number_density')
10  .filterDate('2019-04-01', '2019-10-01');
```

## 8) Average the formaldehyde and NO<sub>2</sub> variables.

- a. Create a new variable named “formaldehyde\_mean” to store the averaged pixel values.

- b. Set the variable equal to the original formaldehyde variable and use the `.mean()` command to take the average value of all the days within our date range.
- c. Use the `.clip()` command to clip the image to the geometry of interest.
  - i. In this case, the geometry of interest is the “geometry” variable storing the coordinates.
- d. Repeat steps 8a-8c for NO<sub>2</sub>.

```

FNR Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')
6   .filterDate('2019-04-01', '2019-10-01');
7
8 var no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
9   .select('tropospheric_NO2_column_number_density')
10  .filterDate('2019-04-01', '2019-10-01');
11
12 var formaldehyde_mean = formaldehyde.mean().clip(geometry);
13 var no2_mean = no2.mean().clip(geometry);
14

```

### 9) Take the ratio.

- a. Create a variable named `fnr` to store the ratio data in.
- b. Divide the formaldehyde mean variable by the `no2_mean` variable using the `.divide()` command.
- c. Rename the band to “ratio” to describe more accurately what is now stored in the variable.

```

FNR Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')
6   .filterDate('2019-04-01', '2019-10-01');
7
8 var no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
9   .select('tropospheric_NO2_column_number_density')
10  .filterDate('2019-04-01', '2019-10-01');
11
12 var formaldehyde_mean = formaldehyde.mean().clip(geometry);
13 var no2_mean = no2.mean().clip(geometry);
14
15 var fnr = formaldehyde_mean.divide(no2_mean).rename('ratio');
16

```

### 10) Plot the ratio.

- a. Set up the visualization parameters in the `band_viz` variable.
  - i. In this document, the maximum value is 10, the minimum value is 0, and a rainbow color scheme is created for the FNR.
- b. Add the data to the map using `Map.addLayer(*insert variable name*, *insert visualization parameters*, *insert the name of the layer*)`.
  - i. In this document, we will plot only the FNR but if you are interested in plotting the HCHO or the NO<sub>2</sub> data simply use `formaldehyde_mean` or `no2_mean` as the variable name in the `Map.addLayer()` command.

Additionally, adjust the max value in the visualization parameters to ensure a more aesthetically pleasing display. We recommend starting with maximum values of about 0.0001 for NO<sub>2</sub> and 0.0002 for HCHO.

- c. Click the "Run" button. It may take a couple of minutes for the output (shown in Step 11) to be displayed on the map.

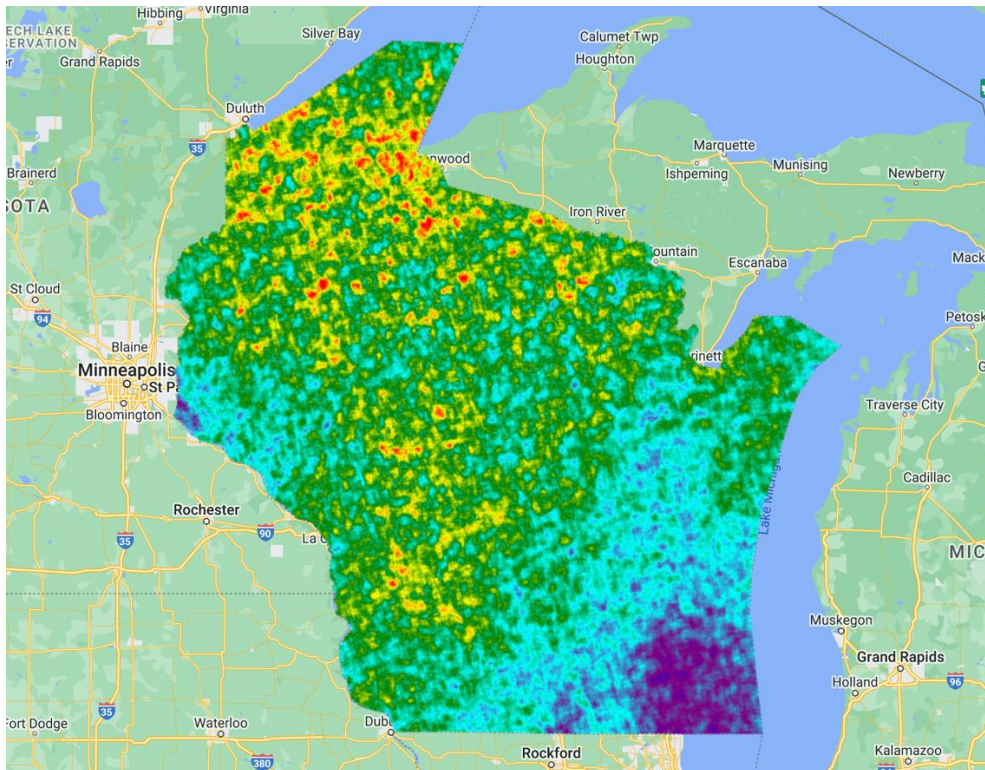
```

FNR
Get Link Save Run Reset
1 var geometry = ee.FeatureCollection("TIGER/2018/States")
2   .filter(ee.Filter.eq('NAME', 'Wisconsin'));
3
4 var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
5   .select('tropospheric_HCHO_column_number_density')
6   .filterDate('2019-04-01', '2019-10-01');
7
8 var no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
9   .select('tropospheric_NO2_column_number_density')
10  .filterDate('2019-04-01', '2019-10-01');
11
12 var formaldehyde_mean = formaldehyde.mean().clip(geometry);
13 var no2_mean = no2.mean().clip(geometry);
14
15 var fnr = formaldehyde_mean.divide(no2_mean).rename('ratio');
16
17 var band_viz = {
18   min: 0,
19   max: 10,
20   palette: ['black', 'blue', 'purple', 'cyan', 'green', 'yellow', 'red']
21 };
22
23 Map.addLayer(fnr, band_viz, 'Formaldehyde/NO2');
24

```

### 11) Output.

- a. The easiest way to save this image to screenshot it. We will go over a more precise method in Tutorial #2



## 12) Create a legend.

a. Copy and paste the following code:

```
//Creating a legend
function makeColorBarParams(palette) {
  return {
    bbox: [0, 0, 1, 0.1],
    dimensions: '100x10',
    format: 'png',
    min: 0,
    max: 1,
    palette: palette,
  };
}

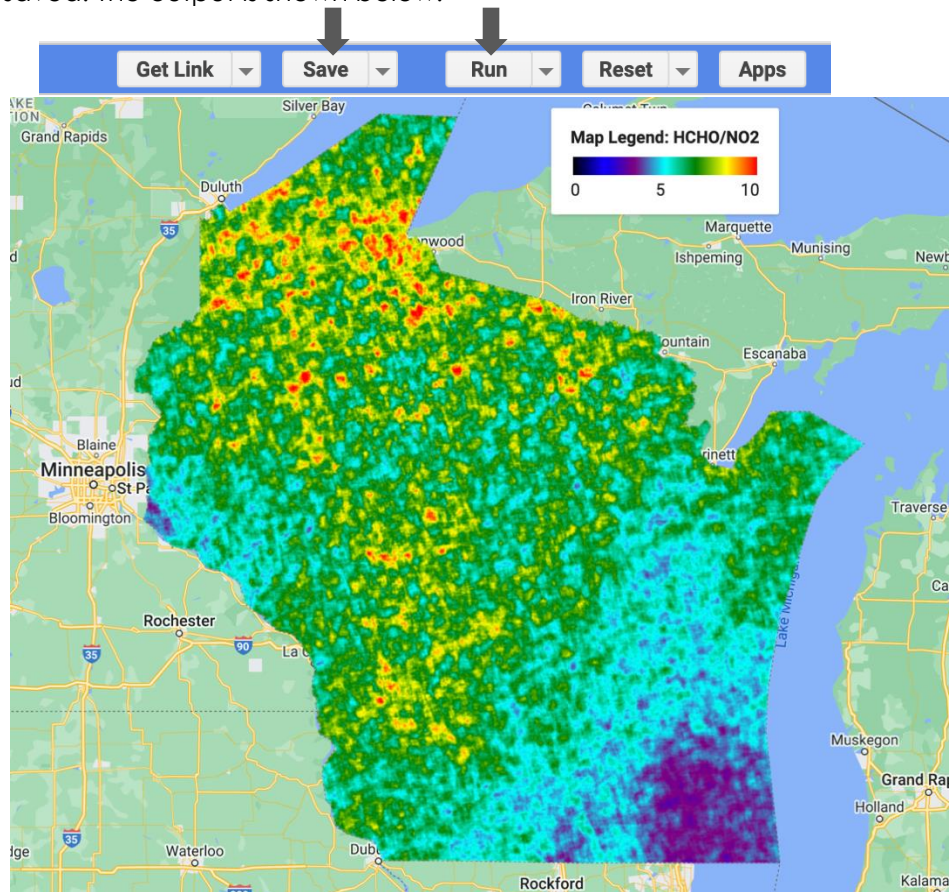
var colorBar = ui.Thumbnail({
  image: ee.Image.pixelLonLat().select(0),
  params: makeColorBarParams(band_viz.palette),
  style: {stretch: 'horizontal', margin: '0px 10px', maxHeight: '24px'},
});

var legendLabels = ui.Panel({
  widgets: [
    ui.Label(band_viz.min, {margin: '4px 8px'}),
    ui.Label(
      ((band_viz.max-band_viz.min) / (2+band_viz.min)),
      {margin: '4px 8px', textAlign: 'center', stretch: 'horizontal'}),
    ui.Label(band_viz.max, {margin: '4px 8px'})
  ],
  layout: ui.Panel.Layout.flow('horizontal')
});

var legendTitle = ui.Label({
  value: 'Map Legend: HCHO/NO2',
  style: {fontWeight: 'bold'}
});

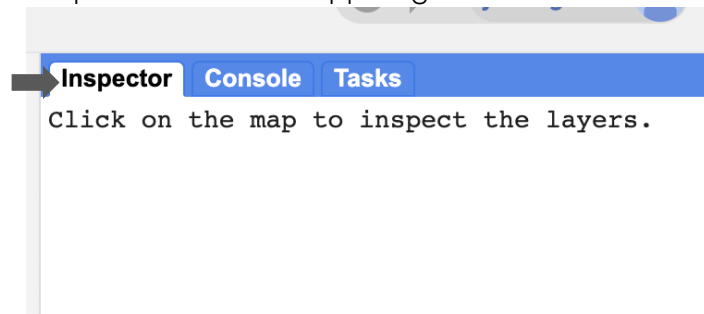
var legendPanel = ui.Panel([legendTitle, colorBar, legendLabels]);
Map.add(legendPanel);
```

- b. Click the “Save” and “Run” button to plot the color bar and ensure all code is saved. The output is shown below.



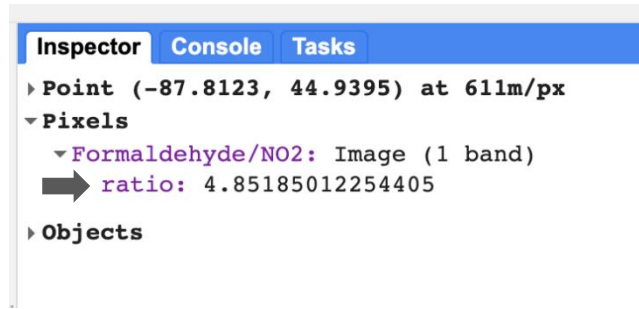
### 13) Determining exact values.

- a. Click on the “Inspector” tab in the upper right corner of the screen.



- b. Click the spot on the map you would like the exact value of the ratio for.

c. The value at that exact pixel will then be shown in the “Inspector” tab.



#### 14) Interpretation of the data

As previously defined, areas with an FNR lower than 3.2 are most likely VOC-limited (suggesting decreases in VOC emissions would most efficiently decrease ozone production) and areas with an FNR higher than 4.1 are most likely NO<sub>x</sub>-limited (suggesting decreases in NO<sub>x</sub> emissions would most efficiently decrease ozone production). Areas with an FNR in between 3.2 and 4.1 are most likely within the transitional regime and a decrease in both VOCs and NO<sub>x</sub> emissions are needed to decrease the production of ozone.

Now we will apply these thresholds to the imagery produced in Step 11. Looking at the southeast corner of Wisconsin, there are some dark blue areas that indicate the area is VOC-limited. This is most likely due to the proximity to Milwaukee, WI and Chicago, IL since cities tend to have more combustion sources and NO<sub>x</sub> emissions, thus making VOCs the limiting component. However, most of the southeast corner is purple indicating a transitional range where a decrease in both VOCs and NO<sub>x</sub> is recommended to decrease ozone production. On the other hand, northern Wisconsin exhibits a higher FNR, with some areas reaching a value of 10. This higher FNR value corresponds with a NO<sub>x</sub>-limited regime. The higher FNR makes sense as northern Wisconsin has an abundance of trees and other natural sources of VOCs, making NO<sub>x</sub> the limiting component.

## • • • • • Conclusion

For another tutorial using Google Earth Engine for ozone sensitivity please see the document labeled “Tutorial 2 Plotting TROPOMI Formaldehyde and Nitrogen Dioxide Satellite Data: An Indicator of Ozone Sensitivity.” The next tutorial will compare the FNR for higher-ozone days to lower-ozone days to assist in identifying the most efficient strategies to decrease ozone concentrations on the most polluted days.

## Code

```
var geometry = ee.FeatureCollection("TIGER/2018/States")
  .filter(ee.Filter.eq('NAME', 'Wisconsin'));

var formaldehyde = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_HCHO')
  .select('tropospheric_HCHO_column_number_density')
  .filterDate('2019-04-01', '2019-10-01');

var no2 = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')
  .select('tropospheric_NO2_column_number_density')
  .filterDate('2019-04-01', '2019-10-01');

var formaldehyde_mean = formaldehyde.mean().clip(geometry);
var no2_mean = no2.mean().clip(geometry);
var fnr = formaldehyde_mean.divide(no2_mean).rename('ratio');

var band_viz = {
  min: 0,
  max: 10,
  palette: ['black', 'blue', 'purple', 'cyan', 'green', 'yellow', 'red']
};

Map.addLayer(fnr, band_viz, 'Formaldehyde/NO2');

//Creating a legend
function makeColorBarParams(palette) {
  return {
    bbox: [0, 0, 1, 0.1],
    dimensions: '100x10',
    format: 'png',
    min: 0,
    max: 1,
    palette: palette,
  };
}
var colorBar = ui.Thumbnail({
  image: ee.Image.pixelLonLat().select(0),
  params: makeColorBarParams(band_viz.palette),
  style: {stretch: 'horizontal', margin: '0px 10px', maxHeight: '24px'},
});
var legendLabels = ui.Panel({
  widgets: [
    ui.Label(band_viz.min, {margin: '4px 8px'}),
    ui.Label(
      ((band_viz.max-band_viz.min) / (2+band_viz.min)),
      {margin: '4px 8px', textAlign: 'center', stretch: 'horizontal'}),
    ui.Label(band_viz.max, {margin: '4px 8px'})
  ],
  layout: ui.Panel.Layout.flow('horizontal')
});
var legendTitle = ui.Label({
  value: 'Map Legend: HCHO/NO2',
  style: {fontWeight: 'bold'}
});
var legendPanel = ui.Panel([legendTitle, colorBar, legendLabels]);
Map.add(legendPanel);
```

## References

Eskes, H., Geffen, J. van, Boersma, F., Eichmann, K.-U., Apituley, A., Pedernana, M., Sneep, M., Veefkind, J. P., & Loyola. (2019). *Sentinel-5 precursor/TROPOMI Level 2 Product User Manual Nitrogen dioxide*. 3.

Jin, X., Fiore, A., Boersma, K. F., Smedt, I. D., & Valin, L. (2020). Inferring Changes in Summertime Surface Ozone–NO<sub>x</sub>–VOC Chemistry over U.S. Urban Areas from Two Decades of Satellite and Ground-Based Observations. *Environmental Science & Technology*, 54(11), 6518–6529. <https://doi.org/10.1021/acs.est.9b07785>

Romahn, F., Pedernana, M., Loyola, D., Apituley, A., Sneep, J., & Veefkind, J. P. (2022). Sentinel-5 precursor/TROPOMI Level 2 Product User Manual Formaldehyde HCHO. *Remote Sensing of Environment*, 120, 70–83. <https://doi.org/10.1016/j.rse.2011.09.027>

Witman, S. & Holloway, Tracey & Reddy, Patrick. (2014). Integrating satellite data into air quality management: Experience from Colorado. *EM: Air and Waste Management Association's Magazine for Environmental Managers*. 34-38.