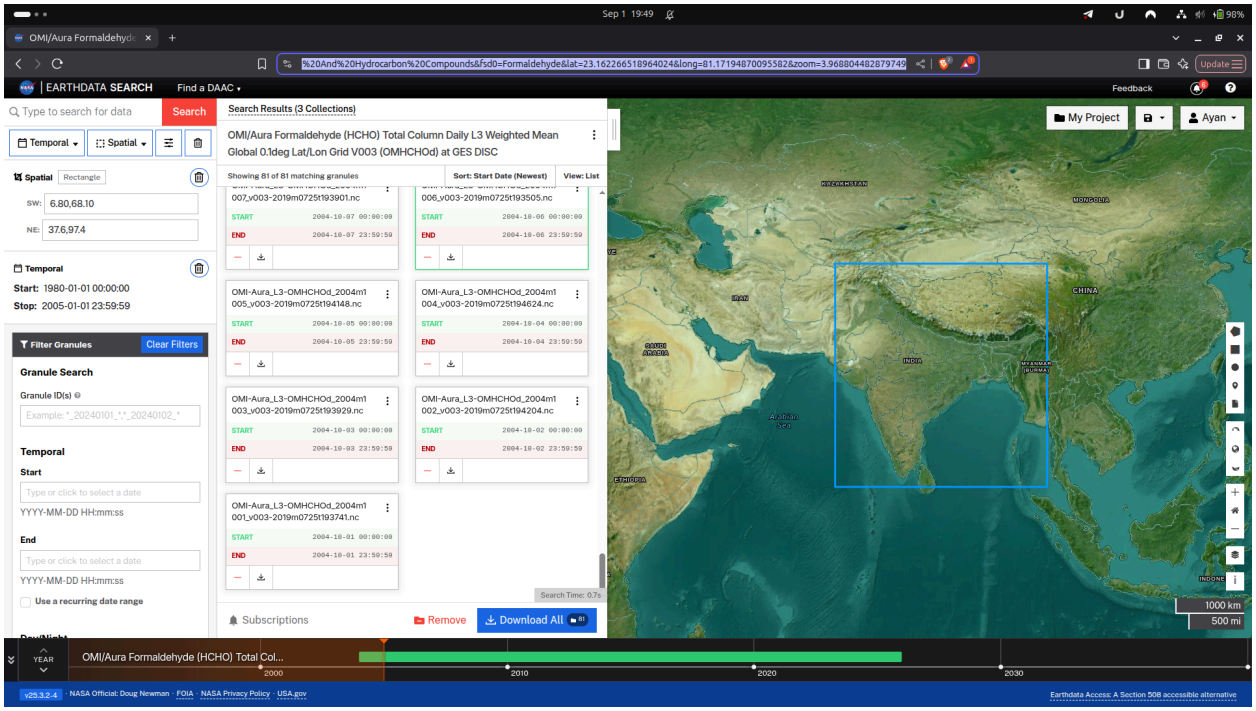
Data scource

Data From the Earthdata NASA website

Link-
https://search.earthdata.nasa.gov/search/granules?p=C1626121562-GES_DISC!C1626121562-GES_DISC&pg[1][v]=t&pg[1][gsk]=-start_date&pg[1][m]=download&pg[1][cd]=f&sb[0]=68.10%2C6.80%2C97.4%2C37.6&qt=1980-01-01T00%3A00%3A00.000Z%2C2005-01-01T23%3A59%3A59.999Z&tl=1426662477.49!5!!&fst0=Atmosphere&fsm0=Atmospheric%20Chemistry&fs10=Carbon%20And%20Hydrocarbon%20Compounds&fsd0=Formaldehyde&lat=23.162266518964024&long=81.17194870095582&zoom=3.968804482879749



Data of location

trial and error

File	Size (MB)	Last Modified	Min	Max	Mean	Valid Cells
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.nc	63.02	2025-08-25 14:58:58	-9.96E+018	6.01E+018	3340000000000000	2216094 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.nc	64.77	2025-08-25 15:00:38	-9.9E+018	1.15E+018	2820000000000000	2304462 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0729_v003-2025m0730t065246.nc	64.68	2025-08-25 15:02:32	-9.83E+018	7.09E+018	3920000000000000	2274916 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0730_v003-2025m0731t075819.nc	62.57	2025-08-25 15:04:27	-9.95E+018	6.83E+017	4360000000000000	2193019 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0731_v003-2025m0801t063054.nc	63.08	2025-08-25 15:15:05	-9.89E+018	1.41E+018	3670000000000000	2222459 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t071938.nc	64.28	2025-08-22 15:19:32	-9.51E+018	7.54E+018	2640000000000000	2262211 / 6480000

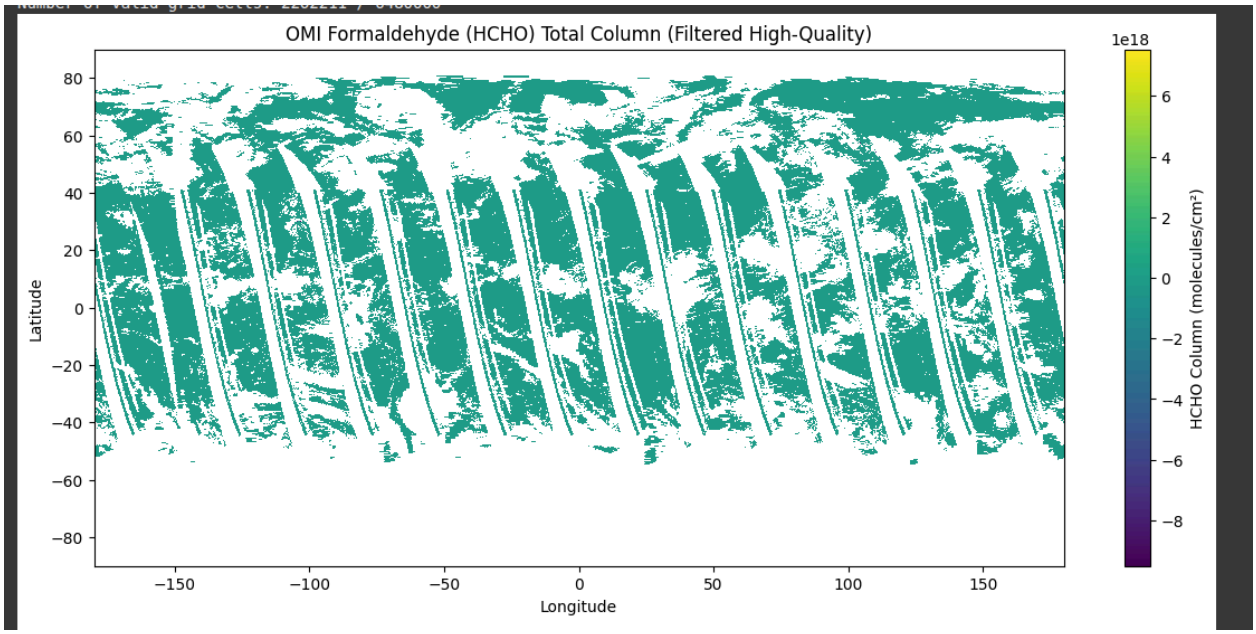
What we can do from each file

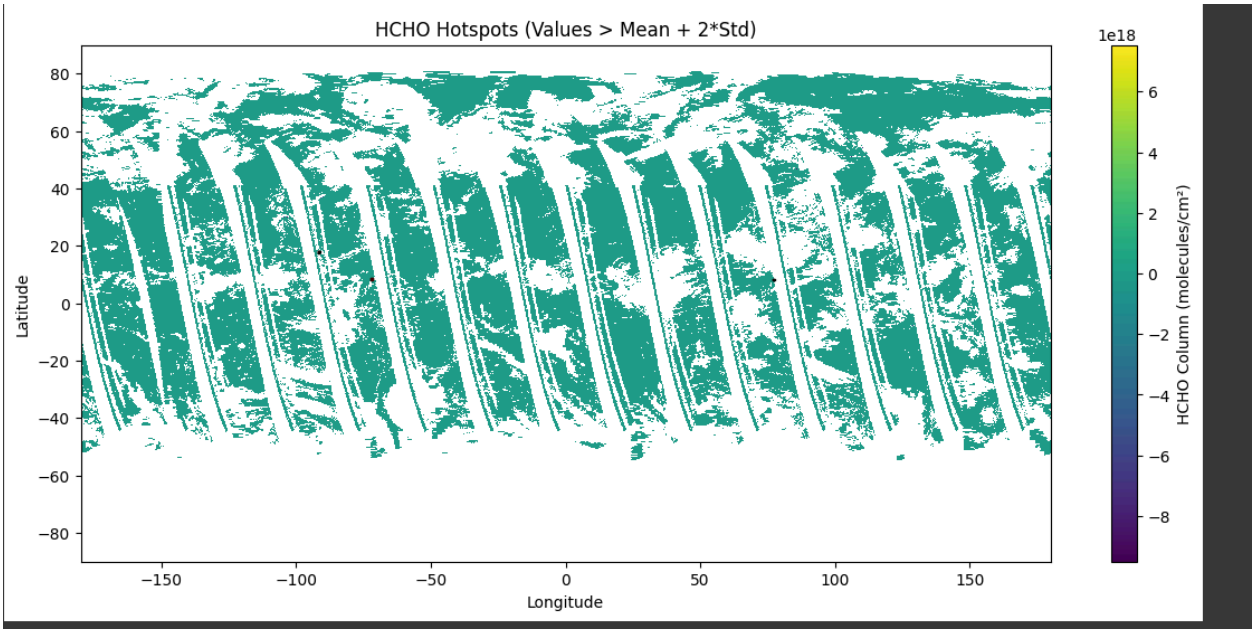
Eg -
https://drive.google.com/file/d/1vSyHo0HLmsHilyH9jANWdR5tK5dRelYq/view?usp=drive_link

📥 Enter the Google Drive shareable link of the NetCDF file:
https://drive.google.com/file/d/1vSyHo0HLmsHilyH9jANWdR5tK5dRelYq/view?usp=drive_link
⬇️ Downloading file from Google Drive...
Downloading...
From: https://drive.google.com/uc?id=1vSyHo0HLmsHilyH9jANWdR5tK5dRelYq
To: /content/OMI_HCHO.nc
100%|[redacted]| 67.4M/67.4M [00:01<00:00, 35.1MB/s]
📂 File path: OMI_HCHO.nc
ℹ️ Size: 64.28 MB
📅 Last modified: 2025-08-22 15:19:32

📊 HCHO Column Summary (Filtered):
Min: -9.50628e+18
Max: 7.53609e+18
Mean: 2.64013e+15
Number of valid grid cells: 2262211 / 6480000

This was the data of the whole world showing an instance suing heatmap





What other data we can extract from each .nc file (all the meatdeatas)

```
📊 HCHO Column Summary:
Min: -9.54134e+18, Max: 7.53609e+18, Mean: 1.60334e+15
Uncertainty: Min 1.21679e+15, Max 8.50235e+18, Mean 1.02068e+16
Number of samples: Min 1.00003e-06, Max 3.97699, Mean 0.549759
Quality flag unique values: [0 1 --]
Cloud fraction: Min 0, Max 0.3, Mean 0.111238
```

- `All the meta data whcih we have` Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
- Saving OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t071938.nc to OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t071938.nc
-
- 📂 Uploaded file: OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t071938.nc
- ℹ️ Size: 64.28 MB
- 📅 Last modified: 2025-08-22 15:38:56.877599
-
- ✅ NetCDF Format: NETCDF4
- 📄 File format: NETCDF4
-
- 🌍 Global Attributes:
- • ShortName: OMHCHOd
- • LongName: OMI/Aura Formaldehyde (HCHO) Total Column Daily L3 Weighted Mean Global 0.1deg Lat/Lon Grid

- ・title: OMI/Aura Formaldehyde (HCHO) Total Column Daily L3 Weighted Mean Global 0.1deg Lat/Lon Grid
- ・VersionID: 003
- ・Format: NetCDF-4
- ・ProcessingLevel: Level 3
- ・IdentifierProductionDOIAuthority: https://www.doi.org
- ・IdentifierProductDOI: 10.5067/AURA/OMI/DATA3010
- ・Conventions: CF-1.7
- ・source: OMI
- ・DataSetQuality: OMHCHOd dataset has been computed after excluding level 2 observations with SZA > 70 and cloud fraction > 0.3 and including only the highest quality retrievals as characterized by the level 2 product quality flags. Further details can be found in the OMHCHOd.README file
- ・DayOfYear: 231
- ・references: Sun et al., 2018 (https://doi.org/10.5194/amt-11-6679-2018),Gonzalez Abad et al., 2015 (https://doi.org/10.5194/amt-8-19-2015)
- ・ContactPersonName: Kelly Chance
- ・ContactPersonRole: Lead algorithm scientist
- ・ContactPersonEmail: kchange@cfa.harvard.edu
- ・institution: OMI SIPS
- ・GranuleID: OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t071938.nc
- ・ProductionDateTime: 2025m0820t071938Z
- ・history: 2025m0820t071938Z; Original file created by OMI SIPS
- ・RangeBeginningDate: 2025-08-19
- ・RangeBeginningTime: 00:00:00Z
- ・RangeEndingDate: 2025-08-20
- ・RangeEndingTime: 00:00:00Z
- ・ObservationArea: Global
- ・InputOriginalFile: OMI-Aura_L2-OMHCHO_2025m0818t2353-o112215_v003-2025m0819t062538.he5,OMI-Aura_L2-OMHCHO_2025m0819t0132-o112216_v003-2025m0819t075428.he5,OMI-Aura_L2-OMHCHO_2025m0819t0311-o112217_v003-2025m0819t100055.he5,OMI-Aura_L2-OMHCHO_2025m0819t0449-o112218_v003-2025m0819t124439.he5,OMI-Aura_L2-OMHCHO_2025m0819t0628-o112219_v003-2025m0819t145042.he5,OMI-Aura_L2-OMHCHO_2025m0819t0807-o112220_v003-2025m0819t145426.he5,OMI-Aura_L2-OMHCHO_2025m0819t0945-o112221_v003-2025m0819t162147.he5,OMI-Aura_L2-OMHCHO_2025m0819t1124-o112222_v003-2025m0819t175009.he5,OMI-Aura_L2-OMHCHO_2025m0819t1303-o112223_v003-2025m0819t210517.he5,OMI-Aura_L2-OMHCHO_2025m0819t1441-o112224_v003-2025m0819t223852.he5,OMI-Aura_L2-OMHCHO_2025m0819t1620-o112225_v003-2025m0819t223928.he5,OMI-Aura_L2-OMHCHO_2025m0819t1759-o112226_v003-2025m0820t003343.he5,OMI-Aura_L2-OMHCHO_2025m0819t1937-o112227_v003-2025m0820t020015.he5,OMI-Aura_L2-OMHCHO_2025m0819t2116-o112228_v003-2025m0820t052627.he5,OMI-Aura_L2-OMHCHO_2025m0819t2254-o112229_v003-2025m0820t065942.he5
- ・ProductGenerationAlgorithm: OMHCHOd

- • ProductGenerationAlgorithmVersion: 0.0.10
- • SouthernmostLatitude: -90.0
- • NorthernmostLatitude: 90.0
- • WesternmostLongitude: -180.0
- • EasternmostLongitude: 180.0
- • LatitudeResolution: 0.10000000149011612
- • LongitudeResolution: 0.10000000149011612
- • MaximumCloudFraction: 0.30000001192092896
- • MaximumSZA: 70.0
-
- 📏 Dimensions:
- • longitude: 3600
- • latitude: 1800
-
- 📊 Variables:
-
- 🔹 longitude
- Shape: (3600,)
- Type: float32
- Min: -179.95, Max: 179.972, Mean: 0.0109852
-
- 🔹 latitude
- Shape: (1800,)
- Type: float32
- Min: -89.95, Max: 89.9473, Mean: -0.00137153
-
- 📁 Groups:
- • Group: key_science_data
- Path: /key_science_data
- Dimensions: []
- Variables: ['column_amount', 'column_uncertainty']
- • Group: qa_statistics
- Path: /qa_statistics
- Dimensions: []
- Variables: ['num_samples', 'data_quality_flag']
- • Group: support_data
- Path: /support_data
- Dimensions: []
- Variables: ['albedo', 'terrain_height', 'amf', 'cloud_fraction', 'cloud_pressure', 'sample_weight']

Effective data which we can use for HCHO
Summarized csv file

File	Size_MB	Last_Modified	HCHO_Min	HCHO_Max	HCHO_Mean	Valid_Cells
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	63.02	2025-08-25 14:58:58	-9.96E+018	6.01E+018	33400000000000000	2216094 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	64.77	2025-08-25 15:00:38	-9.9E+018	1.15E+018	28200000000000000	2304462 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0729_v003-2025m0730t065246.	64.68	2025-08-25 15:02:32	-9.83E+018	7.09E+018	39200000000000000	2274916 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0730_v003-2025m0731t075819.	62.57	2025-08-25 15:04:27	-9.95E+018	6.83E+017	43600000000000000	2193019 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0731_v003-2025m0801t063054.	63.08	2025-08-25 15:15:05	-9.89E+018	1.41E+018	36700000000000000	2222459 / 6480000
OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t071938.	64.28	2025-08-22 15:19:32	-9.51E+018	7.54E+018	26400000000000000	2262211 / 6480000

Other relevent deta points

File	Group	Variable	Dimensions	Shape	Units	Description
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	root	longitude	('longitude',)	(3600,)	degrees_east	longitude
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	root	latitude	('latitude',)	(1800,)	degrees_north	latitude
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	key_science_data	column_amount	('latitude', 'longitude')	(1800, 3600)	molecules/cm^2	HCHO column amount
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	key_science_data	column_uncertainty	('latitude', 'longitude')	(1800, 3600)	molecules/cm^2	HCHO column uncertainty
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	qa_statistics	num_samples	('latitude', 'longitude')	(1800, 3600)	1	number of samples
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	qa_statistics	data_quality_flag	('latitude', 'longitude')	(1800, 3600)		main data quality flag
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	support_data	albedo	('latitude', 'longitude')	(1800, 3600)	1	surface albedo
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	support_data	terrain_height	('latitude', 'longitude')	(1800, 3600)	m	terrain height
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	support_data	amf	('latitude', 'longitude')	(1800, 3600)	1	air mass factor
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	support_data	cloud_fraction	('latitude', 'longitude')	(1800, 3600)	1	AMF cloud fraction
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	support_data	cloud_pressure	('latitude', 'longitude')	(1800, 3600)	hPa	AMF cloud pressure
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.	support_data	sample_weight	('latitude', 'longitude')	(1800, 3600)	1	sample weight
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	root	longitude	('longitude',)	(3600,)	degrees_east	longitude
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	root	latitude	('latitude',)	(1800,)	degrees_north	latitude
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	key_science_data	column_amount	('latitude', 'longitude')	(1800, 3600)	molecules/cm^2	HCHO column amount
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	key_science_data	column_uncertainty	('latitude', 'longitude')	(1800, 3600)	molecules/cm^2	HCHO column uncertainty
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	qa_statistics	num_samples	('latitude', 'longitude')	(1800, 3600)	1	number of samples
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	qa_statistics	data_quality_flag	('latitude', 'longitude')	(1800, 3600)		main data quality flag
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	support_data	albedo	('latitude', 'longitude')	(1800, 3600)	1	surface albedo
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	support_data	terrain_height	('latitude', 'longitude')	(1800, 3600)	m	terrain height
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	support_data	amf	('latitude', 'longitude')	(1800, 3600)	1	air mass factor
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	support_data	cloud_fraction	('latitude', 'longitude')	(1800, 3600)	1	AMF cloud fraction
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	support_data	cloud_pressure	('latitude', 'longitude')	(1800, 3600)	hPa	AMF cloud pressure
OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t074949.	support_data	sample_weight	('latitude', 'longitude')	(1800, 3600)	1	sample weight

GPT Explanation of what each parameter are :-

🌍 Root Variables

- **longitude** (degrees_east)
  → The longitude values (from -180° to +180°), defining the east-west grid of the Earth.

- **latitude** (degrees_north)
  → The latitude values (from -90° to +90°), defining the north-south

grid of the Earth.

🔑 Group: key_science_data (main product variables)

These are the **core scientific measurements**.

- **column_amount** (molecules/cm²)
  → The total **formaldehyde (HCHO) column amount** in the atmosphere, measured as the number of molecules per square centimeter of air column.
  👉 Think of it as: *"How much HCHO is there in a vertical column of air, above each grid cell?"*

- **column_uncertainty** (molecules/cm²)
  → The estimated **uncertainty** (error margin) for the HCHO column measurement.
  👉 Higher uncertainty = less reliable measurement.

📊 Group: qa_statistics (quality assurance info)

These are used to **filter out bad data**.

- **num_samples** (1)
  → Number of satellite retrievals (measurements) that were averaged into this grid cell.
  👉 More samples = higher reliability.

- **data_quality_flag** (1)
  → A flag that indicates whether the data for this grid cell is "good" or "bad."

  - 0 = good

- 1 or higher = lower quality (often excluded)

🔧 Group: `support_data` (extra geophysical + retrieval info)

These variables help explain conditions that affect the measurement.

- **albedo** (1, unitless)
  → Surface reflectivity (fraction of sunlight reflected by the ground).
  👉 Bright surfaces (like deserts, snow) have high albedo and influence the satellite retrieval.

- **terrain_height** (m)
  → Surface elevation above sea level at that grid point.

- **amf** (1, unitless, "air mass factor")
  → Relates the measured slant column (what the satellite sees at an angle) to the vertical column.
  👉 A correction factor depending on sunlight path length and scattering.

- **cloud_fraction** (1, unitless)
  → Fraction of the grid cell covered by clouds (0 = clear, 1 = fully cloudy).
  👉 Retrievals with high cloud fraction are usually discarded.

- **cloud_pressure** (hPa)
  → Pressure level of the cloud top (helps to locate cloud altitude).

- **sample_weight** (1, unitless)
  → A weighting factor for averaging grid cells, accounts for varying reliability of each measurement.

🗒️ Quick "Cheat Sheet" (what matters most)

Variable	Meaning	Why Important
column_amount	Formaldehyde column density	Main data (target pollutant HCHO)
column_uncertainty	Error estimate	Tells reliability
num_samples	Number of retrievals	More = better average
data_quality_flag	Good/bad flag	Used for filtering
cloud_fraction	% of clouds	Filter out cloudy data
cloud_pressure	Cloud top pressure	Retrieval context
albedo	Surface reflectivity	Retrieval accuracy
terrain_height	Surface elevation	Contextual info
amf	Air mass factor	Conversion correction
sample_weight	Statistical weight	For averaging multiple points

Your files contain **Level 3** daily gridded data for formaldehyde measurements:

File naming convention breakdown:
OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t074353.nc

- OMI-Aura: Instrument and satellite
- L3: Data processing level (gridded, daily averages)
- OMHCHOd: HCHO daily product
- 2025m0727: Date (July 27, 2025)
- v003: Product version
- Processing timestamp information

Data Structure and Variables

Coordinate System

- **Longitude**: 3,600 points covering -180° to +180° (0.1° resolution)
- **Latitude**: 1,800 points covering -90° to +90° (0.1° resolution)
- **Grid**: Global coverage with approximately 11 km × 11 km pixels at equator

Key Science Variables

Primary Measurements

- **column_amount**: The main measurement - total HCHO molecules in a vertical column through the atmosphere (molecules/cm²)
- **column_uncertainty**: Measurement uncertainty/error estimates (molecules/cm²)

Quality Assessment

- **num_samples**: Number of individual satellite observations averaged into each grid cell
- **data_quality_flag**: Quality indicators for each measurement
- **sample_weight**: Statistical weights used in averaging process

Supporting Information

- **albedo**: Surface reflectivity (affects measurement accuracy)
- **terrain_height**: Ground elevation (meters above sea level)

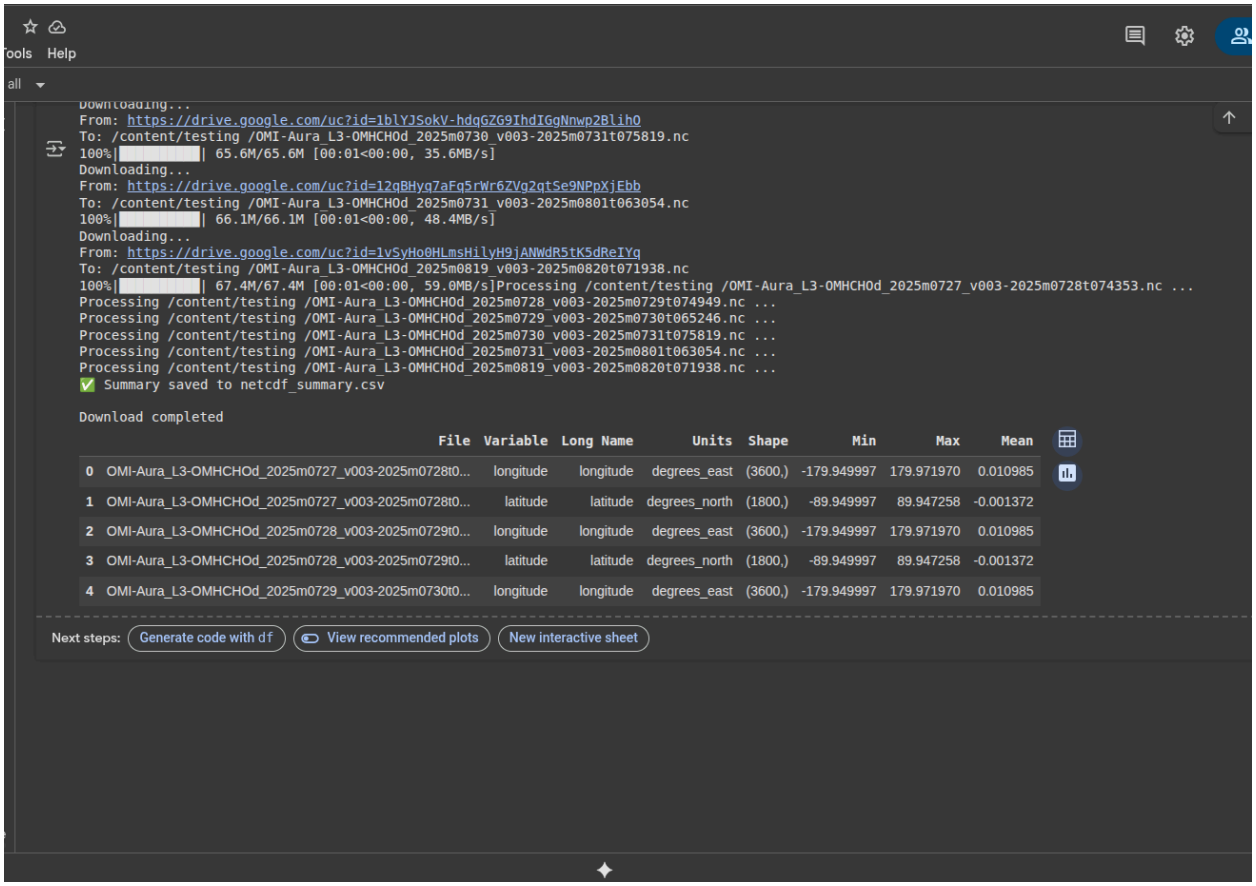
- **amf**: Air Mass Factor (correction for atmospheric path length)
- **cloud_fraction**: Fraction of pixel covered by clouds
- **cloud_pressure**: Atmospheric pressure at cloud top (hPa)

Data Processing Levels

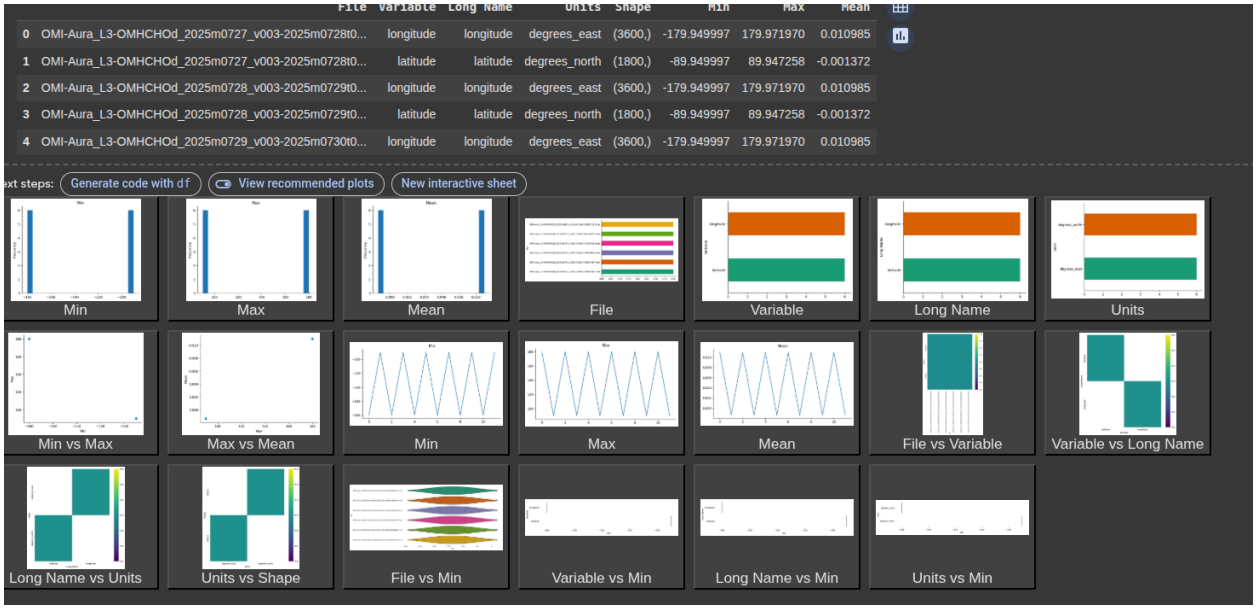
Your data is **Level 3**, which means:

- **Level 0**: Raw instrument data
- **Level 1**: Calibrated radiances
- **Level 2**: Geophysical parameters (individual satellite pixels)
- **Level 3**: **Your data** - Gridded, temporally averaged products
- **Level 4**: Model assimilated or higher-level derived products

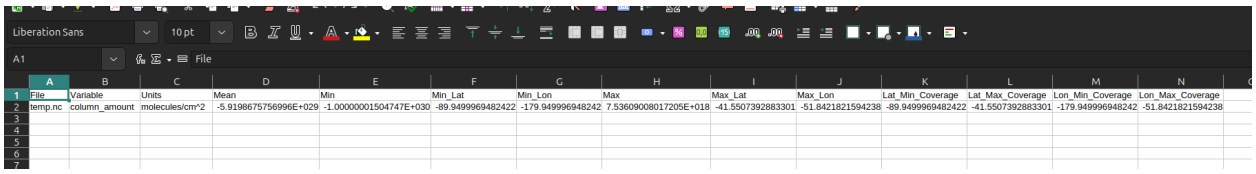
_



	File	Variable	Long Name	Units	Shape	Min	Max	Mean
0	OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t0...	longitude	longitude	degrees_east	(3600,)	-179.949997	179.971970	0.010985
1	OMI-Aura_L3-OMHCHOd_2025m0727_v003-2025m0728t0...	latitude	latitude	degrees_north	(1800,)	-89.949997	89.947258	-0.001372
2	OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t0...	longitude	longitude	degrees_east	(3600,)	-179.949997	179.971970	0.010985
3	OMI-Aura_L3-OMHCHOd_2025m0728_v003-2025m0729t0...	latitude	latitude	degrees_north	(1800,)	-89.949997	89.947258	-0.001372
4	OMI-Aura_L3-OMHCHOd_2025m0729_v003-2025m0730t0...	longitude	longitude	degrees_east	(3600,)	-179.949997	179.971970	0.010985



Data in each .nc file



File	Variable	Units	Mean	Min	Min_Lat	Min_Lon	Max	Max_Lat	Max_Lon	Lat_Min_Coverage	Lat_Max_Coverage	Lon_Min_Coverage	Lon_Max_Coverage
temp.nc	column_amount	molecules/cm^2	-5.9198675756996E+029	-1.00000001504747E+030	-89.9499969482422	-179.949996948242	7.53609008017205E+018	-41.5507392883301	-51.8421821594238	-89.9499969482422	-41.5507392883301	-179.949996948242	-51.8421821594238

Inspecting file structure:

Global Attributes

ShortName: OMHCHOd

LongName: OMI/Aura Formaldehyde (HCHO) Total Column Daily L3 Weighted Mean Global 0.1deg Lat/Lon Grid

title: OMI/Aura Formaldehyde (HCHO) Total Column Daily L3 Weighted Mean Global 0.1deg Lat/Lon Grid

VersionID: 003

Format: NetCDF-4

ProcessingLevel: Level 3

IdentifierProductionDOIAuthority: https://www.doi.org

IdentifierProductDOI: 10.5067/AURA/OMI/DATA3010

Conventions: CF-1.7

source: OMI

DataSetQuality: OMHCHOd dataset has been computed after excluding level 2 observations with SZA > 70 and cloud fraction > 0.3 and including only the highest quality retrievals as characterized by the level 2 product quality flags. Further details can be found in the OMHCHOd.README file

DayOfYear: 231

references: Sun et al., 2018 (https://doi.org/10.5194/amt-11-6679-2018),Gonzalez Abad et al., 2015 (https://doi.org/10.5194/amt-8-19-2015)

ContactPersonName: Kelly Chance

ContactPersonRole: Lead algorithm scientist

ContactPersonEmail: kchange@cfa.harvard.edu

institution: OMI SIPS

GranuleID: OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t071938.nc

ProductionDateTime: 2025m0820t071938Z

history: 2025m0820t071938Z; Original file created by OMI SIPS

RangeBeginningDate: 2025-08-19

RangeBeginningTime: 00:00:00Z

RangeEndingDate: 2025-08-20

RangeEndingTime: 00:00:00Z

ObservationArea: Global

InputOriginalFile: OMI-Aura_L2-OMHCHO_2025m0818t2353-o112215_v003-2025m0819t062538.he5,OMI-Aura_L2-OMHCHO_2025m0819t0132-o112216_v003-2025m0819t075428.he5,OMI-Aura_L2-OMHCHO_2025m0819t0311-o112217_v003-2025m0819t100055.he5,OMI-Aura_L2-OMHCHO_2025m0819t0449-o112218_v003-2025m0819t124439.he5,OMI-Aura_L2-OMHCHO_2025m0819t0628-o112219_v003-2025m0819t145042.he5,OMI-Aura_L2-OMHCHO_2025m0819t0807-o112220_v003-2025m0819t145426.he5,OMI-Aura_L2-OMHCHO_2025m0819t0945-o112221_v003-2025m0819t162147.he5,OMI-Aura_L2-OMHCHO_2025m0819t1124-o112222_v003-2025m0819t175009.he5,OMI-Aura_L2-OMHCHO_2025m0819t1303-o112223_v003-2025m0819t210517.he5,OMI-Aura_L2-OMHCHO_2025m0819t1441-o112224_v003-2025m0819t223852.he5,OMI-Aura_L2-OMHCHO_2025m0819t1620-o112225_v003-2025m0819t223928.he5,OMI-Aura_L2-OMHCHO_2025m0819t1759-o112226_v003-2025m0820t003343.he5,OMI-Aura_L2-OMHCHO_2025m0819t1937-o112227_v003-2025m0820t020015.he5,OMI-Aura_L2-OMHCHO_2025m0819t2116-o112228_v003-2025m0820t052627.he5,OMI-Aura_L2-OMHCHO_2025m0819t2254-o112229_v003-2025m0820t065942.he5

ProductGenerationAlgorithm: OMHCHOd

ProductGenerationAlgorithmVersion: 0.0.10

SouthernmostLatitude: -90.0

NorthernmostLatitude: 90.0

WesternmostLongitude: -180.0

EasternmostLongitude: 180.0

LatitudeResolution: 0.10000000149011612

LongitudeResolution: 0.10000000149011612

MaximumCloudFraction: 0.30000001192092896

MaximumSZA: 70.0

Variables

longitude — dims: ('longitude',), shape: (3600,), units: degrees_east

latitude — dims: ('latitude',), shape: (1800,), units: degrees_north

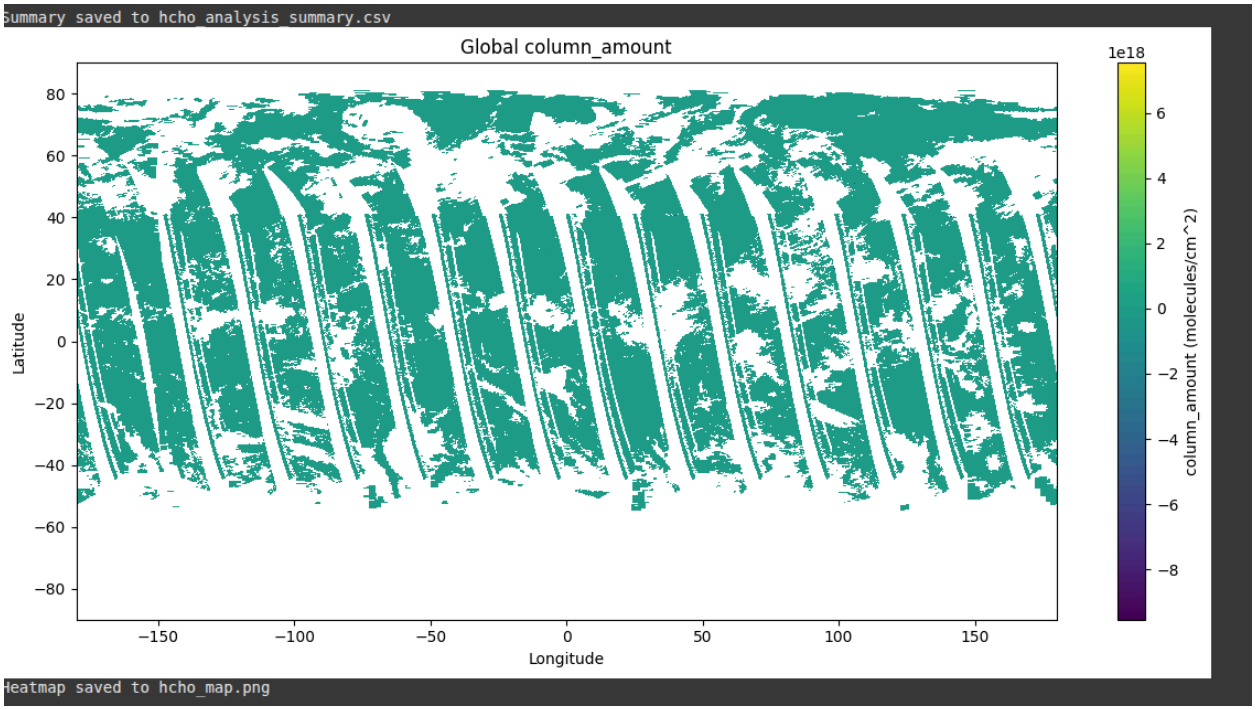
```
Inspecting file structure:
### Global Attributes
ShortName                                                                OMHCHOd
LongName                          OMI/Aura Formaldehyde (HCHO) Total Column Dail...
title                             OMI/Aura Formaldehyde (HCHO) Total Column Dail...
VersionID                                                                    003
Format                                                                   NetCDF-4
ProcessingLevel                                                           Level 3
IdentifierProductionDOIAuthority                                https://www.doi.org
IdentifierProductDOI                                      10.5067/AURA/OMI/DATA3010
Conventions                                                              CF-1.7
source                                                                      OMI
DataSetQuality                    OMHCHOd dataset has been computed after exclud...
DayOfYear                                                                   231
references                        Sun et al., 2018 (https://doi.org/10.5194/amt-...
ContactPersonName                                                    Kelly Chance
ContactPersonRole                                      Lead algorithm scientist
ContactPersonEmail                                          kchange@cfa.harvard.edu
institution                                                           OMI SIPS
GranuleID                         OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t0...
ProductionDateTime                                          2025m0820t071938Z
history                           2025m0820t071938Z; Original file created by OM...
RangeBeginningDate                                                2025-08-19
RangeBeginningTime                                                00:00:00Z
RangeEndingDate                                                   2025-08-20
RangeEndingTime                                                   00:00:00Z
ObservationArea                                                      Global
InputOriginalFile                 OMI-Aura_L2-OMHCHO_2025m0818t2353-o112215_v003...
ProductGenerationAlgorithm                                              OMHCHOd
ProductGenerationAlgorithmVersion                                       0.0.10
SouthernmostLatitude                                                   -90.0
NorthernmostLatitude                                                    90.0
WesternmostLongitude                                                  -180.0
EasternmostLongitude                                                   180.0
LatitudeResolution                                                       0.1
LongitudeResolution                                                      0.1
MaximumCloudFraction                                                     0.3
MaximumSZA                                                              70.0
dtype: object
```

Variables

	Variable	Dimensions	Shape	Units
0	longitude	(longitude,)	(3600,)	degrees_east
1	latitude	(latitude,)	(1800,)	degrees_north

Generating summary table...

	File	Variable	Units	Mean	Min	Max
0	temp.nc	longitude	degrees_east	0.010985	-179.949997	179.971970
1	temp.nc	latitude	degrees_north	-0.001372	-89.949997	89.947258



	A	B	C	D	E	F
1	File	Variable	Units	Mean	Min	Max
2	temp.nc	longitude	degrees_east	0.0109852431342006	-179.949996948242	179.971969604492
3	temp.nc	latitude	degrees_north	-0.00137152778916061	-89.9499969482422	89.9472579956055

- https://www.giss.nasa.gov/tools/panoply/
- Net cdf file processing tool.
-

gpt says

OMI-Aura HCHO Satellite Data Documentation

Table of Contents

Overview

The **OMI-Aura Level 3 Formaldehyde (HCHO) Total Column Daily Dataset** provides global measurements of atmospheric formaldehyde concentrations from the Ozone Monitoring Instrument (OMI) aboard NASA's Aura satellite. This documentation covers the structure, variables, and usage of these NetCDF4 format files.

[INSERT IMAGE: OMI-Aura satellite illustration or diagram]

Key Features

- **Global Coverage**: Complete Earth coverage at 0.1° × 0.1° resolution
- **Daily Temporal Resolution**: One file per day
- **High Quality**: Pre-filtered for optimal data quality
- **Multiple Variables**: Column amounts, uncertainties, and supporting geophysical data

[INSERT IMAGE: Sample global HCHO column density map]

Dataset Description

What is Formaldehyde (HCHO)?

Formaldehyde is a trace gas in the atmosphere that serves as an important indicator of:

- **Air pollution** from industrial sources and vehicles
- **Biogenic emissions** from vegetation and natural processes
- **Biomass burning** from wildfires and agricultural fires
- **Secondary formation** from hydrocarbon oxidation

[INSERT DIAGRAM: HCHO sources and atmospheric processes]

Dataset Specifications

- **Product Name**: OMHCHOd (OMI HCHO daily)
- **Spatial Resolution**: 0.1° × 0.1° (approximately 11 km × 11 km at equator)
- **Temporal Resolution**: Daily
- **Coverage**: Global (-90° to +90° latitude, -180° to +180° longitude)
- **File Format**: NetCDF4
- **Data Level**: Level 3 (gridded, processed)

File Structure and Metadata

File Naming Convention

OMI-Aura_L3-OMHCHOd_YYYYmMMdd_v003-YYYYmMMddtHHmmss.nc

Example: `OMI-Aura_L3-OMHCHOd_2025m0819_v003-2025m0820t071938.nc`

- Date: August 19, 2025
- Version: 003
- Processing timestamp: August 20, 2025 at 07:19:38 UTC

Sample Dataset Statistics

File Date	Size (MB)	Valid Cells	Coverage
2025-07-27	63.02	2,216,094 / 6,480,000	34.2%
2025-07-28	64.77	2,304,462 / 6,480,000	35.6%

2025-07-29	64.68	2,274,916 / 6,480,000	35.1%
2025-07-30	62.57	2,193,019 / 6,480,000	33.8%
2025-07-31	63.08	2,222,459 / 6,480,000	34.3%
2025-08-19	64.28	2,262,211 / 6,480,000	34.9%

[INSERT CHART: Coverage statistics over time]

Global Attributes

The files contain comprehensive metadata including:

- **Contact Information**: Kelly Chance (kchange@cfa.harvard.edu), Lead algorithm scientist
- **Data Quality**: Pre-filtered for SZA < 70°, cloud fraction < 0.3
- **References**: Sun et al., 2018; Gonzalez Abad et al., 2015
- **Processing**: OMI SIPS (Science Investigator-led Processing System)

Data Variables Reference

Grid Coordinates

`longitude` (degrees_east)

- **Shape**: (3600,)
- **Range**: -179.95° to +179.972°
- **Resolution**: 0.1°
- **Description**: East-west grid coordinates

`latitude` (degrees_north)

- **Shape**: (1800,)
- **Range**: -89.95° to +89.947°
- **Resolution**: 0.1°
- **Description**: North-south grid coordinates

[INSERT DIAGRAM: Global grid system illustration]

Key Science Data (`/key_science_data/`)

`column_amount` (molecules/cm²)

- **Primary Variable**: ✅ Main scientific measurement
- **Units**: molecules per square centimeter
- **Typical Range**: ~10^{15} to 10^{16} molecules/cm²
- **Description**: Total atmospheric formaldehyde column density
- **Interpretation**: Higher values indicate more HCHO in the vertical air column

[INSERT IMAGE: Global HCHO column amount visualization with color scale]

`column_uncertainty` (molecules/cm²)

- **Units**: molecules per square centimeter
- **Range**: ~10^{15} to 10^{18} molecules/cm²
- **Description**: Estimated measurement uncertainty
- **Usage**: Quality assessment and error propagation in analyses

[INSERT IMAGE: Uncertainty distribution map or histogram]

Quality Assurance (`/qa_statistics/`)

`num_samples` (dimensionless)

- **Range**: ~10^{-6} to ~4
- **Mean**: ~0.55
- **Description**: Number of Level 2 retrievals averaged into each grid cell
- **Quality Indicator**: More samples = higher reliability

`data_quality_flag` (dimensionless)

- **Values**:
  - `0` = Good quality data
  - `1` = Lower quality
  - `--` = Missing data
- **Usage**: Filter data by excluding non-zero flags

[INSERT CHART: Quality flag distribution pie chart]

Support Data (`/support_data/`)

Geophysical Context Variables

`cloud_fraction` (dimensionless)

- **Range**: 0 to 0.3 (pre-filtered)
- **Description**: Fraction of grid cell covered by clouds
- **Impact**: Higher cloud fraction reduces retrieval accuracy

`cloud_pressure` (hPa)

- **Description**: Pressure at cloud top level
- **Usage**: Contextual information for cloud-affected retrievals

`terrain_height` (meters)

- **Description**: Surface elevation above sea level
- **Impact**: High altitude areas may have different HCHO profiles

`albedo` (dimensionless)

- **Range**: 0 to 1
- **Description**: Surface reflectivity
- **Impact**: Affects satellite measurement sensitivity

[INSERT DIAGRAM: How surface properties affect satellite measurements]

Retrieval Parameters

`amf` (dimensionless)

- **Description**: Air Mass Factor - conversion between slant and vertical columns
- **Technical**: Accounts for light path geometry and atmospheric scattering
- **Quality**: Values far from 1.0 may indicate retrieval challenges

`sample_weight` (dimensionless)

- **Description**: Statistical weight for grid cell averaging
- **Usage**: Accounts for varying measurement reliability

Data Quality and Processing

Quality Filters Applied

The Level 3 data has been pre-processed with quality filters:

1. **Solar Zenith Angle**: SZA ≤ 70°
2. **Cloud Screening**: Cloud fraction ≤ 0.3
3. **Quality Flags**: Only highest quality Level 2 retrievals included

[INSERT FLOWCHART: Data processing and quality control workflow]

Data Coverage

- **Typical Coverage**: ~35% of global grid cells have valid data per day
- **Coverage Variation**: Depends on clouds, sun angle, and instrument availability
- **Missing Data**: Represented as fill values in NetCDF

[INSERT MAP: Typical daily coverage pattern]

Recommended Additional Filtering

For research applications, consider additional filters:

```
# Example filtering criteria
good_data = (
    (data_quality_flag == 0) &          # Good quality only
    (num_samples > 0.1) &               # Sufficient samples
    (column_uncertainty < 5e15) &       # Reasonable uncertainty
    (cloud_fraction < 0.2)              # Extra cloud screening
)
```

Working with the Data

File Access Methods

1. Google Drive Integration

Files can be accessed via Google Drive shareable links:

Example: https://drive.google.com/file/d/1vSyHo0HLmsHilyH9jANWdR5tK5dReIYq/view

2. Direct Download

Use tools like `gdown` to download from Google Drive:

```
import gdown
url = 'https://drive.google.com/uc?id=1vSyHo0HLmsHilyH9jANWdR5tK5dReIYq'
gdown.download(url, 'OMI_HCHO.nc', quiet=False)
```

Reading the Data

Python with xarray (Recommended)

```
import xarray as xr
import numpy as np

# Open dataset
ds = xr.open_dataset('OMI_HCHO.nc')

# Access main variable
hcho_column = ds['key_science_data/column_amount']
uncertainty = ds['key_science_data/column_uncertainty']
quality_flag = ds['qa_statistics/data_quality_flag']

# Apply quality filter
good_data = quality_flag == 0
hcho_filtered = hcho_column.where(good_data)
```

[INSERT CODE SCREENSHOT: Example data loading and filtering]

Alternative Tools

- **MATLAB**: Use `ncread()` function
- **IDL**: Use `ncdf_open()` routines
- **R**: Use `ncdf4` package
- **Command Line**: `ncdump`, `ncview` for quick inspection

Data Visualization Examples

Basic Global Map

```
import matplotlib.pyplot as plt
import cartopy.crs as ccrs

fig = plt.figure(figsize=(12, 8))
ax = plt.axes(projection=ccrs.PlateCarree())
ax.coastlines()

# Plot HCHO data
im = ax.pcolormesh(ds.longitude, ds.latitude,
                   hcho_filtered * 1e-15,  # Convert to 10^15 molecules/cm²
                   transform=ccrs.PlateCarree(),
                   cmap='YlOrRd', vmin=0, vmax=10)

plt.colorbar(im, label='HCHO Column (×10¹⁵ molecules/cm²)')
plt.title('OMI HCHO Column Density')
```

[INSERT IMAGE: Example visualization output]

Applications and Use Cases

1. Air Quality Monitoring

- **Urban Pollution**: Track HCHO hotspots over cities
- **Industrial Emissions**: Monitor point sources and industrial areas
- **Trend Analysis**: Multi-year time series for pollution trends

[INSERT CHART: Time series of HCHO over major cities]

2. Biomass Burning Detection

- **Fire Monitoring**: Identify wildfire and agricultural burning events
- **Seasonal Patterns**: Track burning seasons globally
- **Emissions Estimation**: Quantify fire-related HCHO emissions

[INSERT IMAGE: HCHO enhancement during fire events]

3. Biogenic Emissions Research

- **Forest Emissions**: Study natural HCHO from vegetation
- **Seasonal Cycles**: Analyze vegetation emission patterns
- **Climate Impact**: Understand biosphere-atmosphere interactions

4. Model Validation

- **Chemical Transport Models**: Validate HCHO simulations
- **Emission Inventories**: Evaluate bottom-up emission estimates
- **Process Studies**: Test chemical mechanism representations

[INSERT CHART: Model vs. satellite comparison example]

Technical Specifications

Measurement Principle

OMI measures HCHO using differential optical absorption spectroscopy (DOAS) in the UV spectral region (328.5-356.5 nm).

[INSERT DIAGRAM: OMI measurement principle and spectroscopy]

Retrieval Algorithm

- **Version**: 003
- **Reference Spectrum**: Solar irradiance
- **Spectral Fitting**: Polynomial baseline correction
- **Air Mass Factor**: Calculated using atmospheric profiles

Uncertainties and Limitations

- **Random Error**: ~40-100% for individual pixels
- **Systematic Error**: ~30% for column amounts
- **Detection Limit**: ~5×10^{14} molecules/cm² for daily averages
- **Interference**: Possible from other absorbing species

[INSERT TABLE: Detailed uncertainty breakdown by conditions]

Validation Status

The dataset has been validated against:

- Ground-based MAX-DOAS measurements
- Aircraft observations
- Other satellite instruments (TROPOMI, GOME-2)

[INSERT CHART: Validation statistics and correlation plots]

Data Processing Workflow

[INSERT FLOWCHART: Complete data processing from L1 to L3]

Level 1 → Level 2

1. Radiance calibration
2. Spectral fitting (DOAS)
3. Air mass factor calculation
4. Quality flag assignment

Level 2 → Level 3

1. Spatial gridding (0.1° × 0.1°)
2. Temporal averaging (daily)
3. Quality filtering
4. Statistical computation

Best Practices

Data Usage Guidelines

1. **Always apply quality filters** using `data_quality_flag`
2. **Consider uncertainty** for quantitative analysis
3. **Account for missing data** in spatial/temporal averaging
4. **Validate results** against independent measurements when possible
5. **Cite properly** using provided DOI: 10.5067/AURA/OMI/DATA3010

Common Pitfalls

- ❌ Using data without quality filtering

- ❌ Ignoring measurement uncertainties
- ❌ Assuming complete spatial coverage
- ❌ Mixing different algorithm versions
- ❌ Over-interpreting single-day anomalies

[INSERT CHECKLIST: Data quality verification steps]

File Format Details

NetCDF4 Structure

```
Root Group
├── Dimensions: longitude(3600), latitude(1800)
├── Variables: longitude, latitude
├── Group: key_science_data/
│   └── Variables: column_amount, column_uncertainty
├── Group: qa_statistics/
│   └── Variables: num_samples, data_quality_flag
└── Group: support_data/
    └── Variables: albedo, terrain_height, amf,
                   cloud_fraction, cloud_pressure, sample_weight
```

Missing Data Convention

- **Fill Value**: Varies by variable (check `_FillValue` attribute)
- **Valid Range**: Check `valid_min` and `valid_max` attributes
- **Scale Factor**: Some variables may use scale/offset packing

---

Contact Information

Algorithm Team

- **Lead Scientist**: Kelly Chance
- **Email**: kchange@cfa.harvard.edu
- **Institution**: Harvard-Smithsonian Center for Astrophysics

Data Provider

- **Institution**: OMI Science Investigator-led Processing System (SIPS)
- **Website**: https://disc.gsfc.nasa.gov/datasets/OMHCHO_003/summary

References

1. **Sun, K. et al. (2018)**: "Deriving the slit functions from OMI solar observations and its implications for ozone-profile retrieval." *Atmospheric Measurement Techniques*, 11(12), 6679-6695. DOI: 10.5194/amt-11-6679-2018

2. **González Abad, G. et al. (2015)**: "Updated Smithsonian Astrophysical Observatory Ozone Monitoring Instrument (SAO OMI) formaldehyde retrieval." *Atmospheric Measurement Techniques*, 8(1), 19-32. DOI: 10.5194/amt-8-19-2015

Dataset Citation

NASA/GSFC (2025): OMI/Aura Formaldehyde (HCHO) Total Column Daily L3 Weighted Mean Global 0.1deg Lat/Lon Grid V003, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC). DOI: 10.5067/AURA/OMI/DATA3010

Appendix: Variable Summary Table

Variable	Group	Units	Typical Range	Primary Use
`longitude`	Root	degrees_east	-180 to +180	Grid coordinate
`latitude`	Root	degrees_north	-90 to +90	Grid coordinate
`column_amount`	key_science_data	molecules/cm²	10^{15} - 10^{16}	**Main HCHO data**
`column_uncertainty`	key_science_data	molecules/cm²	10^{15} - 10^{18}	Uncertainty estimate

`num_samples`	qa_statistics	dimensionless	0 - 4	Sample count
`data_quality_flag`	qa_statistics	dimensionless	0, 1, missing	**Quality control**
`cloud_fraction`	support_data	dimensionless	0 - 0.3	Cloud screening
`cloud_pressure`	support_data	hPa	Variable	Cloud context
`terrain_height`	support_data	meters	Variable	Surface elevation
`albedo`	support_data	dimensionless	0 - 1	Surface reflectivity
`amf`	support_data	dimensionless	Variable	Retrieval factor
`sample_weight`	support_data	dimensionless	Variable	Averaging weight

[INSERT VISUAL SUMMARY: Infographic of all variables and their relationships]

Last updated: August 2025
Documentation version: 1.0

codes

Downloading Data and processing NC files

```
import os, netCDF4 as nc
import numpy as np, pandas as pd, requests, time
from tqdm import tqdm
from getpass import getpass

# ------------------------------
# Force CPU (safe for Colab free runtimes)
# ------------------------------
xp = np
GPU_AVAILABLE = False
print("ℹ️ Running in CPU mode with NumPy (GPU skipped)")


# ------------------------------
# Upload CSV
# ------------------------------
from google.colab import files
print("📂 Please upload your CSV file containing the 'urls' column...")
uploaded = files.upload()
csv_file = list(uploaded.keys())[0]
print(f"✅ Uploaded file: {csv_file}")


# ------------------------------
# Earthdata .netrc Setup
# ------------------------------
print("\n🔑 Please enter your NASA Earthdata Login credentials")
username = input("Username (default=ayan_7869): ").strip() or "ayan_7869"
password = getpass("Password: ")

# Create ~/.netrc dynamically in Colab
netrc_path = os.path.expanduser("~/.netrc")
with open(netrc_path, "w") as f:
    f.write(f"machine urs.earthdata.nasa.gov login {username} password
{password}\n")
os.chmod(netrc_path, 0o600)
```

```
print(f"✅ Created .netrc at {netrc_path}")


# ------------------------------
# Helpers
# ------------------------------
skipped_rows = []  # global log for skipped/problematic files


def safe_download(session, url, out_path, retries=3, timeout=60):
   """Robust file downloader with retry, timeout, and HTML detection +
progress bar."""
   for attempt in range(retries):
       try:
           with session.get(url, stream=True, allow_redirects=True,
timeout=timeout) as resp:
               if resp.status_code == 200:
                   total = int(resp.headers.get("content-length", 0))
                   # with open(out_path, "wb") as f, tqdm(
                   #     total=total, unit="B", unit_scale=True,
unit_divisor=1024,
                   #     desc=f"⬇️ {os.path.basename(out_path)}",
leave=False
                   # ) as bar:
                   #     for chunk in resp.iter_content(8192):
                   #         if chunk:
                   #             f.write(chunk)
                   #             bar.update(len(chunk))

                   with open(out_path, "wb") as f:
                     for chunk in resp.iter_content(8192):
                         if chunk:
                             f.write(chunk)


                   # sanity check size
                   size = os.path.getsize(out_path)
                   if size < 10_000:
                       with open(out_path, "r", errors="ignore") as fh:
                           head = fh.read(500)
```

```
                            if "<html" in head.lower():
                                reason = "Downloaded HTML (login/error
page)"
                            else:
                                reason = "Downloaded file too small"
                        skipped_rows.append({"File":
os.path.basename(out_path), "Reason": reason})
                        print(f"⚠️ {os.path.basename(out_path)} skipped →
{reason}")
                        os.remove(out_path)
                        return False
                    return True
                else:
                    raise ValueError(f"HTTP {resp.status_code}")
        except Exception as e:
            print(f"⚠️ Download error ({e}), attempt
{attempt+1}/{retries}")
            time.sleep(3)
    skipped_rows.append({"File": os.path.basename(out_path), "Reason":
"Download failed"})
    return False


def summarize_netcdf(file_path, date=None, processing_timestamp=None):
    rows = []
    try:
        with nc.Dataset(file_path, 'r') as ds:
            if "latitude" not in ds.variables or "longitude" not in
ds.variables:
                reason = "Missing latitude/longitude"
                print(f"⚠️ {file_path} skipped → {reason}")
                skipped_rows.append({"File": os.path.basename(file_path),
"Reason": reason})
                return []

            lat, lon = ds.variables["latitude"][:],
ds.variables["longitude"][:]

            if "key_science_data" not in ds.groups:
                reason = "Missing key_science_data group"
```

```
                print(f"⚠️ {file_path} skipped → {reason}")
                skipped_rows.append({"File": os.path.basename(file_path),
"Reason": reason})
                return []

            if "column_amount" not in
ds.groups["key_science_data"].variables:
                reason = "Missing column_amount variable"
                print(f"⚠️ {file_path} skipped → {reason}")
                skipped_rows.append({"File": os.path.basename(file_path),
"Reason": reason})
                return []

            try:
                var =
ds.groups["key_science_data"].variables["column_amount"]
                arr = xp.array(var[:], dtype=float)
                arr[arr < -1e20] = xp.nan
            except MemoryError:
                reason = "MemoryError while loading array"
                print(f"⚠️ {file_path} skipped → {reason}")
                skipped_rows.append({"File": os.path.basename(file_path),
"Reason": reason})
                return []

            if xp.isnan(arr).all():
                reason = "All values NaN"
                print(f"⚠️ {file_path} skipped → {reason}")
                skipped_rows.append({"File": os.path.basename(file_path),
"Reason": reason})
                return []

            # Stats
            mean_val = float(np.nanmean(arr))
            min_val  = float(np.nanmin(arr))
            max_val  = float(np.nanmax(arr))

            try:
                idx_min = int(np.nanargmin(arr))
                idx_max = int(np.nanargmax(arr))
```

```
                coords_min, coords_max = np.unravel_index(idx_min,
arr.shape), np.unravel_index(idx_max, arr.shape)
                min_lat, min_lon = float(lat[coords_min[0]]),
float(lon[coords_min[1]])
                max_lat, max_lon = float(lat[coords_max[0]]),
float(lon[coords_max[1]])
            except Exception:
                min_lat = min_lon = max_lat = max_lon = None

            DU_factor = 2.69e16

            rows.append({
                "Dataset": os.path.basename(file_path).split("_")[0],
                "File": os.path.basename(file_path),
                "date": date,
                "processing_timestamp": processing_timestamp,
                "Units": "molecules/cm²",
                "HCHO_Mean (molecules/cm²)": f"{mean_val:.3e}" if not
np.isnan(mean_val) else "NA",
                "HCHO_Min (molecules/cm²)": f"{min_val:.3e}" if not
np.isnan(min_val) else "NA",
                "HCHO_Max (molecules/cm²)": f"{max_val:.3e}" if not
np.isnan(max_val) else "NA",
                "HCHO_Mean (DU)": f"{mean_val/DU_factor:.2f}" if not
np.isnan(mean_val) else "NA",
                "HCHO_Min (DU)": f"{min_val/DU_factor:.2f}" if not
np.isnan(min_val) else "NA",
                "HCHO_Max (DU)": f"{max_val/DU_factor:.2f}" if not
np.isnan(max_val) else "NA",
                "Min_Lat": f"{min_lat:.2f}" if min_lat is not None else
"NA",
                "Min_Lon": f"{min_lon:.2f}" if min_lon is not None else
"NA",
                "Max_Lat": f"{max_lat:.2f}" if max_lat is not None else
"NA",
                "Max_Lon": f"{max_lon:.2f}" if max_lon is not None else
"NA",
            })

    except Exception as e:
```

```
        print(f"⚠️ {file_path} skipped → {e}")
        skipped_rows.append({"File": os.path.basename(file_path), "Reason":
f"Processing error: {e}"})

    return rows


# ------------------------------
# Stream download + process
# ------------------------------
def stream_process(csv_file, output_csv="hcho_analysis.csv",
threshold=30):
    df = pd.read_csv(csv_file)
    if "urls" not in df.columns:
        raise ValueError("CSV must have a column named 'urls'")

    if "date" not in df.columns or "processing_timestamp" not in
df.columns:
        raise ValueError("CSV must have 'date' and 'processing_timestamp'
columns")

    session = requests.Session()
    all_rows = []
    output_dir = "./downloads"
    os.makedirs(output_dir, exist_ok=True)

    for i, row in enumerate(tqdm(df.itertuples(), desc="Processing URLs",
unit="url")):
        url = row.urls
        date = getattr(row, "date", None)
        processing_timestamp = getattr(row, "processing_timestamp", None)

        fname = os.path.basename(url.split("?")[0])
        out_path = os.path.join(output_dir, fname)

        # Download
        if not safe_download(session, url, out_path):
            continue

        # Summarize
```

```
        rows = summarize_netcdf(out_path, date, processing_timestamp)
        all_rows.extend(rows)

        # Cleanup
        try:
            if os.path.exists(out_path):
                os.remove(out_path)
        except Exception as e:
            print(f"⚠️ Could not delete {out_path}: {e}")

        # 👇 log progress like "22/31 URLs processed"
        print(f"✅ Processed {i+1}/{len(df)} → {fname}")

        # Save periodically
        if (i + 1) % threshold == 0 or (i + 1) == len(df):
            try:
                pd.DataFrame(all_rows).to_csv(output_csv, index=False)
                print(f"💾 Saved {len(all_rows)} rows to {output_csv} (up
to file {i+1})")
            except Exception as e:
                print(f"⚠️ Failed saving main CSV: {e}")

            if skipped_rows:
                try:

pd.DataFrame(skipped_rows).to_csv("skipped_files_log.csv", index=False)
                except Exception as e:
                    print(f"⚠️ Failed saving skipped log: {e}")

    print(f"\n✅ Finished. Summary saved to {output_csv}")
    if skipped_rows:
        print(f"⚠️ {len(skipped_rows)} files skipped →
skipped_files_log.csv")


# ------------------------------
# Run
# ------------------------------
stream_process(csv_file, output_csv="hcho_analysis_from_2024_2022.csv",
threshold=50)
```

[redacted]

Data Visualizer

```
import pandas as pd
import matplotlib.pyplot as plt
import os
from google.colab import files
import zipfile

# ------------------------------
# Upload CSV
# ------------------------------
print("📂 Please upload your HCHO summary CSV file...")
uploaded = files.upload()
csv_file = list(uploaded.keys())[0]
print(f"✅ Uploaded file: {csv_file}")

# Create output folder
out_dir = os.path.splitext(csv_file)[0]
os.makedirs(out_dir, exist_ok=True)
print(f"📂 Output folder created: {out_dir}")

# ------------------------------
# Load CSV
# ------------------------------
df = pd.read_csv(csv_file)

# Normalize column names
df.columns = df.columns.str.strip()

# Pick date column
date_col = "date"
df[date_col] = pd.to_datetime(df[date_col], errors="coerce")
df = df.sort_values(date_col)

# Pick HCHO columns (Dobson Units)
mean_col = "HCHO_Mean (DU)"
min_col  = "HCHO_Min (DU)"
```

```
max_col  = "HCHO_Max (DU)"

# Convert to numeric
for col in [mean_col, min_col, max_col]:
   df[col] = pd.to_numeric(df[col], errors="coerce")

# ------------------------------
# Clean Data: remove invalid/extreme values
# ------------------------------
df = df[(df[mean_col] >= 0) & (df[mean_col] <= 5)]
df = df.dropna(subset=[mean_col, min_col, max_col])

print(f"✅ Cleaned dataset: {len(df)} valid rows remain")
print("Value ranges:")
print(df[[mean_col, min_col, max_col]].describe())

# ------------------------------
# 1. Time Series with Min-Max Shading
# ------------------------------
plt.figure(figsize=(12,6))
plt.plot(df[date_col], df[mean_col], label="Mean HCHO (DU)", color="blue")
plt.fill_between(df[date_col], df[min_col], df[max_col],
                 color="skyblue", alpha=0.3, label="Min-Max Range")

plt.title("Daily Formaldehyde (HCHO) Column Amount (Cleaned)")
plt.xlabel("Date")
plt.ylabel("HCHO (Dobson Units)")
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(out_dir, "time_series.png"), dpi=300,
bbox_inches="tight")
plt.close()

# ------------------------------
# 2. Histogram
# ------------------------------
plt.figure(figsize=(8,5))
plt.hist(df[mean_col].dropna(), bins=30, color="green", alpha=0.7)
plt.title("Distribution of Daily Mean HCHO (Cleaned)")
plt.xlabel("HCHO (Dobson Units)")
```

```
plt.ylabel("Frequency")
plt.grid(True)
plt.savefig(os.path.join(out_dir, "histogram.png"), dpi=300,
bbox_inches="tight")
plt.close()

# ------------------------------
# 3. Rolling Average (7-day)
# ------------------------------
df["HCHO_7day_MA"] = df[mean_col].rolling(window=7).mean()

plt.figure(figsize=(12,6))
plt.plot(df[date_col], df[mean_col], label="Daily Mean", alpha=0.5)
plt.plot(df[date_col], df["HCHO_7day_MA"], label="7-day Moving Avg",
color="red", linewidth=2)

plt.title("Smoothed Trend of HCHO (7-day Moving Avg, Cleaned)")
plt.xlabel("Date")
plt.ylabel("HCHO (Dobson Units)")
plt.legend()
plt.grid(True)
plt.savefig(os.path.join(out_dir, "rolling_average.png"), dpi=300,
bbox_inches="tight")
plt.close()

# ------------------------------
# Save cleaned CSV
# ------------------------------
cleaned_csv = os.path.join(out_dir, "HCHO_cleaned.csv")
df.to_csv(cleaned_csv, index=False)

# ------------------------------
# Zip & Download
# ------------------------------
zip_filename = f"{out_dir}.zip"
with zipfile.ZipFile(zip_filename, 'w') as zipf:
   for root, _, files_list in os.walk(out_dir):
       for f in files_list:
           zipf.write(os.path.join(root, f),
arcname=os.path.join(os.path.basename(root), f))
```

```
files.download(zip_filename)
```

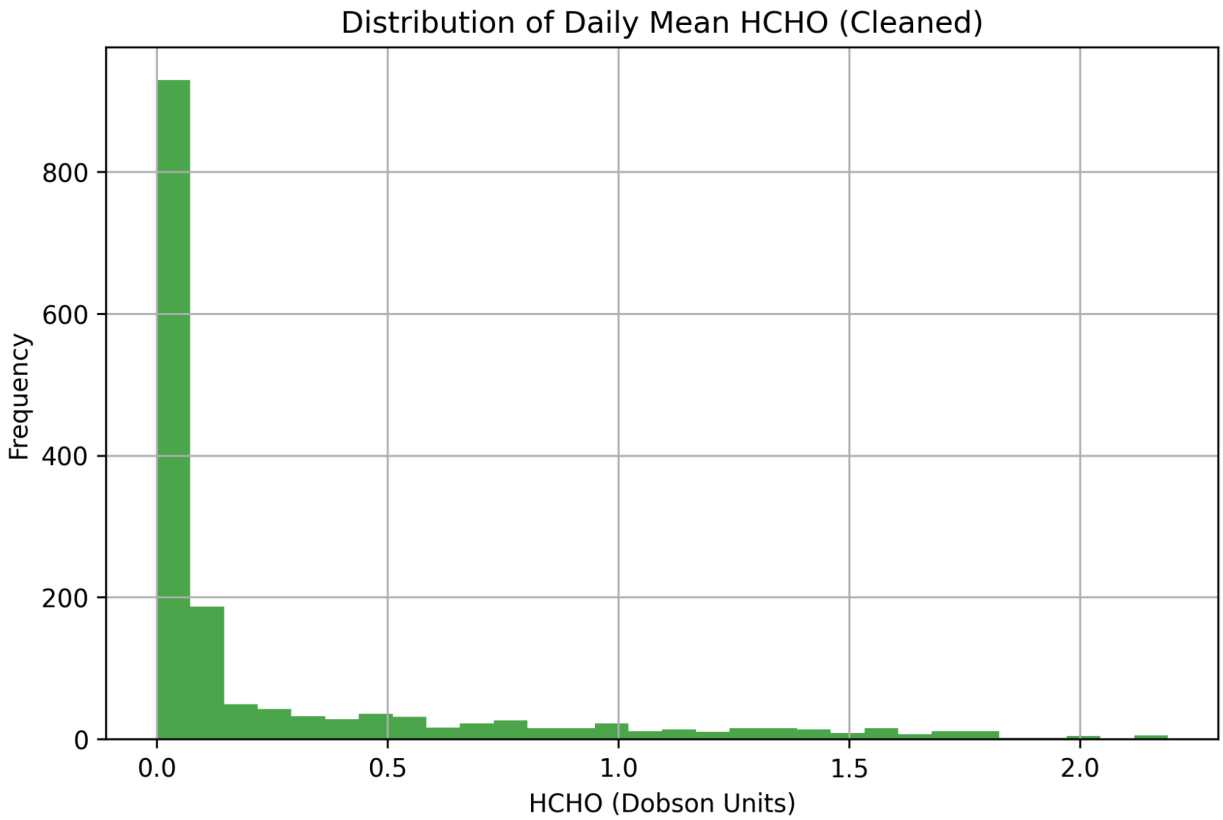
Viz outputs

What does `2.153e+16 molecules/cm^2` mean?

- Your dataset (OMI-Aura L3 OMHCHOd) is reporting **column density**.

- **Column density** = the total number of HCHO molecules in a column of air extending from the surface up to the top of the atmosphere, per unit area on the ground.

- Dobson Unit, a unit of measurement for the total atmospheric ozone concentration
- One Dobson Unit (DU) is equivalent to approximately 2.69 x 10^{16} molecules per square centimeter of a substance.

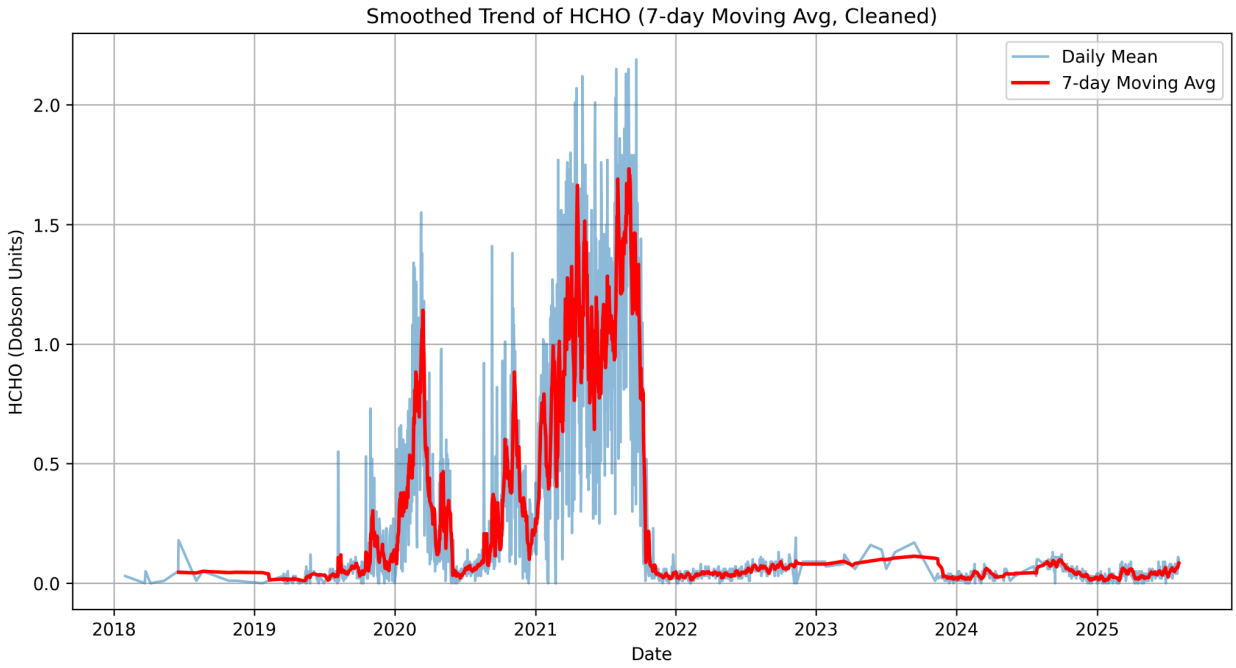


What it shows:

- This plot counts **how many days** had a certain range of **HCHO Mean (DU)** values.
- Most days fall between **0 and 0.1 DU** → which makes sense, because background atmospheric HCHO is usually very small.
- A smaller number of days show higher values (0.5–2 DU), which may be linked to **special events** like crop burning, wildfires, or industrial emissions.

Takeaway:

- HCHO distribution is very **skewed** → mostly low, with a long tail of high values.
- Those few “spikes” are the **events worth investigating**.



What it shows:

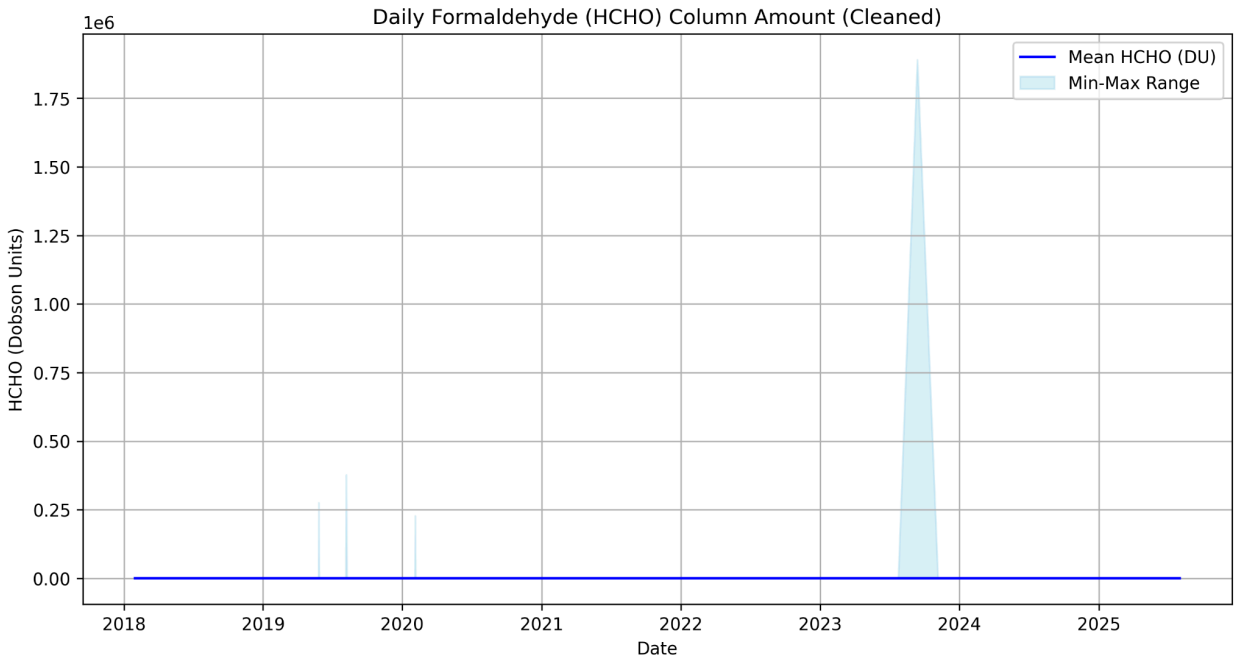
- The **light blue line** = daily mean HCHO over time (2018–2025).
- The **red line** = 7-day moving average (smoother trend).

Observations:

- Between **2019 and 2021**, there were big HCHO increases (up to ~2 DU).
- After 2022, the values dropped back closer to **background levels** (~0.05 DU).
- Spikes align with **seasonal/episodic pollution events** (biomass burning, seasonal vegetation, industrial activity).

Takeaway:

- HCHO had **major pollution episodes around 2020–2021**, but not much after.
- Useful for studying **long-term pollution changes**.



What it shows:

- The **blue line** = mean HCHO (DU).
- The **shaded band** = min–max values for each day.

Problem here:

- The Y-axis goes up to **1e6 DU**, which is unrealistic.
- That means your **HCHO_Min / HCHO_Max (DU)** column still has some corrupted values (e.g., -9999 placeholders).
- The **mean values look fine** (close to 0–2 DU), but min/max aren't cleaned properly yet.

Takeaway:

- This plot is supposed to show "variability" around the mean.
- But currently, the shading is dominated by **bad values**, not real data.
- Once we filter `HCHO_Min/Max` properly (e.g., only keep `0–5 DU` like we did for mean), this plot will look normal.

What else plots can be done

1. Trend & Seasonality Analysis

- **Monthly/Seasonal averages** → aggregate data by month or season (e.g., summer vs. winter HCHO levels).
- **Year-over-Year trends** → detect long-term changes in formaldehyde emissions (possibly linked to pollution control policies, wildfires, or vegetation cycles).
- Use **time-series decomposition** (`statsmodels`) to separate trend, seasonal cycle, and residuals.

2. Spatial Analysis

- Since your dataset has `Min_Lat`, `Max_Lat`, `Min_Lon`, `Max_Lon`, you can:
    - Plot **HCHO concentration maps** (heatmaps or contours) over the region.
    - Use **Cartopy / Basemap / Folium / Plotly maps** in Python to visualize spatial variation.
    - Compare **urban vs. rural** regions.

3. Extreme Events Detection

- Identify days with **spikes** in HCHO (above 95th percentile).
- Correlate spikes with **wildfire data**, **crop burning periods** in India, or **industrial activity**.
- Detect **outliers/anomalies** with machine learning (Isolation Forest, DBSCAN).

4. Correlation with Other Environmental Data

- Compare with:
    - **NO_2** (from OMI satellite) → both are tracers of combustion.
    - **Temperature, rainfall, wind speed** (from ERA5 or NASA MERRA reanalysis).
    - **PM2.5 ground data** from CPCB (India) to see how HCHO links to surface air quality.

5. Machine Learning & Forecasting

- Build models to **predict HCHO levels** from weather + other pollutants.
- Time-series forecasting:
    - ARIMA / SARIMA
    - Facebook Prophet
    - LSTMs/Transformers (deep learning)
- Cluster analysis to group regions/days with similar pollution signatures.

6. Policy & Health Impact Studies

- Compare **pre-lockdown vs. post-lockdown** HCHO levels (e.g., COVID-19 2020).
- Assess impact of **crop residue burning** in Punjab/Haryana on regional HCHO.
- Relate HCHO to **ozone formation potential** (since it's a precursor to tropospheric ozone).

7. Data Products & Visual Dashboards

- Build an **interactive dashboard** in:
    - **Streamlit/Dash** (Python)
    - **Tableau/PowerBI** (business-style reports)
- Show daily/weekly air quality updates.
- Could be valuable for **citizen science projects** or **policy monitoring**.

- Video dashboard link

  https://drive.google.com/file/d/1ArzvHxndMaUXlOeJF2ZfGmlVNxPF4P7A/view?usp=sharing

Tab 6

Random forest output for next year predictions all over india