

# ECMWF

ESoWC

## Air Quality Quality Control Analysis

By Gordon Rates of AirNode Ltd

### User Manual

### Milestone 1

### **Contents**

Aims

Description

User Manual

Dependencies

# Aims

- 1 Import OpenAQ dataset from API source i.e. through the pyOpenAQ API
- 2 Import OpenAQ dataset from bulk download i.e. through ndjson download from AWS server

## Description

Validating and removing errors outliers from surface air quality observations from individual sensors so that these observation can be compared to ECMWF's CAMS air quality forecasts. By clustering analysis on these observations more reliable observations can be identified. Enhancing these observations by attaching data about factors that affect air quality these observations can have more credibility about their accuracy. CAMS lacks credible surface air quality observations in many parts of the world, often in the most polluted area such as in India or Africa. Some observations are available for these areas from data harvesting efforts such as openAQ but there is no quality control applied to the data, and it is often not well known if the observations are made in a rural, urban or heavily polluted local environment. This information on the environment is important because the very locally influenced measurements are mostly not representative for the horizontal scale (40 km) of the CAMS forecasts and should therefore not be used for the evaluation of the CAMS model.

Completed for ECMWF European Centre for Medium Weather Forecast 2020 and ESoWC 2020

## User Manual

### **Aim 1 Import OpenAQ dataset from API source i.e. through the pyOpenAQ API**

#### **Aim 1 Step 1**

- 1 Get the dependencies. These are identified below.
- 2 Make sure the OpenAQ api initiates

#### **Aim 1 Steps 2 - 6 Choose Selection Features**

- 1 Import latest measurements for chosen OpenAQ stations

## **Aim 1 Step 2 Choose OpenAQ download by selecting:**

### **One OpenAQ Station**

**Process: "Milestone1\_Import\_API\_OpenAQStation\_OpenAQ.py"**

1 Input Parameter: OpenAQ Station

Other parameters can be chosen if change from the default

(other parameter defaults: )

Method: Gets the Measurements for chosen OpenAQ station.

2 Output: Spreadsheet in format Comma separated value (CSV) for One OpenAQ Station can be inputted to visual analytics and Quality Control i.e. Milestone 2, 3, 4

**Or**

## **Aim 1 Step 2 Choose OpenAQ Stations download by selecting:**

### **Many OpenAQ Stations by choosing Coordinate Lat, Lng Centre with Radius**

**Process: "Milestone1\_Import\_API\_CoordinateRadius\_OpenAQ.py"**

1 Input: Coordinate Lat, Lng for Centre and Radius

Other parameters can be chosen if change from the default

(other parameter defaults: )

Method: Finds the OpenAQ station within the Radius. It gets the chosen measurements for these OpenAQ stations

2 Output: Separate Spreadsheets or CSV's for the chosen OpenAQ Stations

**Or**

## By Country code from OpenAQ

**Process:** "Milestone1\_Import\_API\_Country\_OpenAQ.py"

1 Input: Country code

Other parameters can be chosen if change from the default

(other parameter defaults: )

Method: Finds the OpenAQ station within the Radius. It gets the chosen measurements for these OpenAQ stations

2 Output: Separate Spreadsheets or CSV's for the chosen OpenAQ Stations

### Aim 1 Step 3 Choose parameter

1 Choose the parameters to import from the OpenAQ stations i.e. 'pm25', 'pm10', 'No2', 'o3' etc from these <https://api.openaq.org/v1/parameters>

```
{"id": "bc", "name": "BC", "description": "Black Carbon", "preferredUnit": "µg/m³"},  
{"id": "co", "name": "CO", "description": "Carbon Monoxide", "preferredUnit": "ppm"},  
{"id": "no2", "name": "NO2", "description": "Nitrogen Dioxide", "preferredUnit": "ppm"},  
{"id": "o3", "name": "O3", "description": "Ozone", "preferredUnit": "ppm"},  
{"id": "pm10", "name": "PM10", "description": "Particulate matter less than 10 micrometers in diameter", "preferredUnit": "µg/m³"},  
{"id": "pm25", "name": "PM2.5", "description": "Particulate matter less than 2.5 micrometers in diameter", "preferredUnit": "µg/m³"},  
{"id": "so2", "name": "SO2", "description": "Sulfur Dioxide", "preferredUnit": "ppm"}
```

### Aim 1 Step 4 Choose 3 and 4 Time Schedule

1 Choose date begin and date end i.e (2018,1,20)

### Aim 1 Step 5 Get the measurements

1 Find OpenAQ Stations

## **Aim 1 Step 6**

1 Choose the iteration of the OpenAQ download

2 Export to CSV to apply

## **Aim 2 Import OpenAQ dataset from bulk download i.e. through ndjson download from AWS server**

### **Step 1**

1 Get dependencies

## **Aim 2 Step 2 - 6 Select OpenAQ Stations from bulk download**

## **Dependencies**

Numpy

Pandas import json\_normalize

datetime

openAQ (Not required for Aim 2)

<http://dhhagan.github.io/py-openaq/tutorial/api.html>

CSV