C-HACK 2022 – PROBLEM 2

General instructions

Note: Required submission and documents are highlighted in red

- Take time to meet with your team and decide how you plan to solve the problem. Define your subtasks and decide who will take care of which subtask and when you will meet to update. Note that **teamwork will be evaluated** at the end of the hackathon.
- Work with your team to **generate a single Jupyter notebook** containing your solution to the given problem. You are free to include additional files needed to run your solution in the same folder, but we should be able to repeat your evaluation simply by running your notebook. If your notebook requires files/modules that are not turned in, we won't be able to run it!
- Your solution should be written in Python 3.
- Make sure that your notebook runs on Google Colab in your Google Drive team folder and is well-documented (i.e. provide explanatory and clarifying comments throughout your code).
- (**REQUIRED**) Write a **1 page summary** which describes your work and your results. You will be evaluated on your summary. The summary should include a title, an introduction, a paragraph which describes your results and a conclusion which summarizes your findings. See the template in the shared google drive folder **C-HACK 2022 EVENT/Templates/**
- Send us pictures of your team and/or your work as you hack your problem! We will be happy to post these on your team webpage and the C-HACK website news feed.
- (**REQUIRED**) Upload your **notebook, all files it depends on and a one page summary of your work** to the google drive folder we have created and shared with your team (this folder will not be seen by any other participants) **by 11:59 PM PDT on Wednesday January 12th 2021.**
- (**REQUIRED**) Prepare a powerpoint or google drive presentation with your team (see the template in **C-HACK 2022 EVENT/Templates/**). **Submit the final version by uploading your presentation** to the shared google drive folder **by 12pm (noon) on Friday January 14th.** Recall that we will be posting your final presentations on the C-HACK website.
- Participate in the dry run on Thursday January 13th to practice presenting with your team.
- (**REQUIRED**) Present your final work with your team (**max 10min** to present) at the Final Presentations and Award ceremony on January 14th.
- Review and follow the code of conduct, which is also posted on the c-hack webpage - https://www.c-hack.org/

Problem 2: Description

As announced this morning, DOW has shared *.csv format files containing data that was gathered from a rectangular waste treatment basin using a drone. In the data files, the positions at which the drone took samples within the basin (x, y position) at three depths (z position) are described by integers. For each sample there are a number of measurements taken: abundance data for different species of bacteria (labeled with “C”), concentrations of different metals (labeled with “FM”), and basin output variables (labeled with “O”). A more detailed description of the data is given below in the provided data section. You can find the files in the shared Google Drive folder: **C-HACK 2021 EVENT/Projects/DOW_data.** Note that you will only be able to view the data. To work with it you must **copy them to your team’s Google Drive directory.**

IMPORTANT: **Do not share the DOW data** with anyone outside the hackathon, this data is proprietary to DOW and has kindly been provided **for the sole purpose of the hackathon**.

In this problem you will work on ***clustering*** this dataset.

Tasks for problem 2

- Evaluate how the distribution of bacterial abundance (labels “C”) changes throughout the basin.
    a. Is the distribution homogeneous in space?
    b. Can you find well defined groups of bacteria (hint: you could use clustering) at a taxonomic level which you believe is most relevant?

- Are your observations on the distribution of bacteria present within the basin **associated with nutrient concentrations** (labels “FM” and “DM”)? Is it possible to predict which bacteria will be present based on nutrient availability?

As you may have noted, the tasks are described in a broad context. We leave you the flexibility to decide…

- At what taxonomic level/s (Kingdom, Genus, …, Species) will you consider the data?
- Biological relationships between species can be complex; how will you identify patterns in bacterial abundance if present?
- How will you identify relationships between nutrients and bacteria?

Provided data and code

Dow Dataset

Copy the contents of the "`DOW_data`" folder to your team's folder!

The dataset comes in two csv files: CHACK2022_abundanceData.csv, and CHACK2022_OutputsMetals.csv. In both files, the position of each sample is in the columns 'location x', 'location y', and 'location z'. For example (x,y,z) = 1,1,1 means basin section 1,1 at depth 1. Please see Figure 1 for a visualization of basin positions. In the first file, bacterial abundance data at various taxonomic levels is provided as columns, with labels starting with "C". This dataframe is multi indexed, which means that a column is not determined by a single label, but multiple. In this case, one column represents a specific species and is labeled with its Kingdom, Phylum, Class, …, species. Please see Figure 2 for how the columns of your multi indexed dataframe look, and Figure 3 for a stylized visualization of the taxonomic levels in the data. In the second file, additional measurements are provided as columns. The concentrations of various metals appear as labels starting with "FM" and "DM" (the metal identities have been hidden for proprietary reasons). Note that "FM" are the fixed metal concentrations, which represent the amount of metal that has been consumed by the bacteria, and "DM" are dissolved metals, so the concentration of metal dissolved in the solution. Other measurements starting with label "O" are described in the table below.

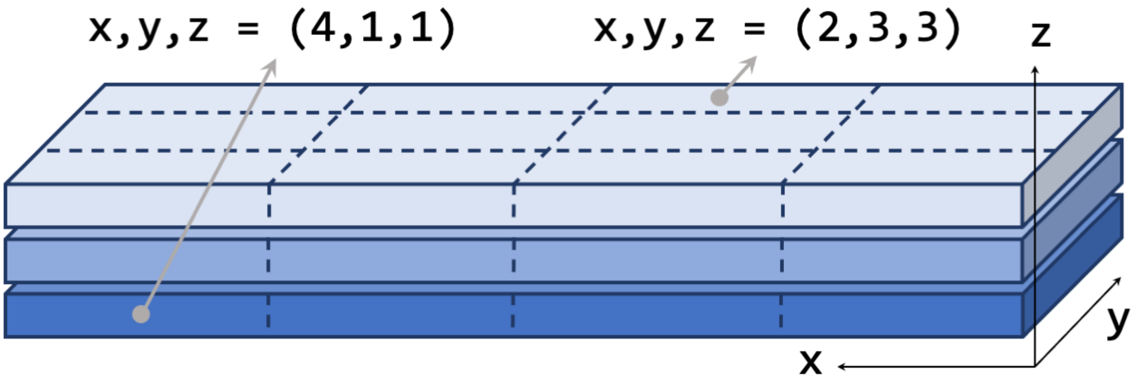
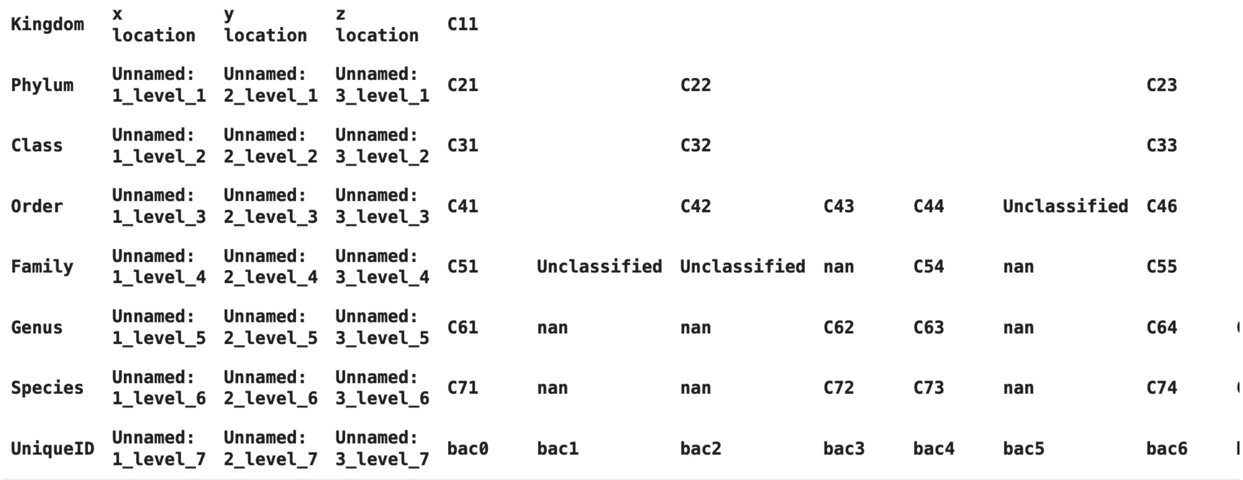


Figure 1: *Illustration of how x, y, z positions are defined in the basin. A set of x, y, z defines one sector of the basin at a certain depth. Two examples are shown. Position 1,1,1 corresponds to the bottom right corner (the origin).*

Figure 2: *The first few column labels of the multi indexed dataframe. The taxonomic levels are shown on the left. The next three columns are the x,y,z positions. The remaining columns specify the classes of Kingdom, Phylum, …, Species that a specific bacteria belongs to. Higher level class covers all lower levels to the right of it. For example, Orders C42, C43, and C44 are*



Kingdom	x location	y location	z location	C11						
Phylum	Unnamed: 1_level_1	Unnamed: 2_level_1	Unnamed: 3_level_1	C21		C22				C23
Class	Unnamed: 1_level_2	Unnamed: 2_level_2	Unnamed: 3_level_2	C31		C32				C33
Order	Unnamed: 1_level_3	Unnamed: 2_level_3	Unnamed: 3_level_3	C41		C42	C43	C44	Unclassified	C46
Family	Unnamed: 1_level_4	Unnamed: 2_level_4	Unnamed: 3_level_4	C51	Unclassified	Unclassified	nan	C54	nan	C55
Genus	Unnamed: 1_level_5	Unnamed: 2_level_5	Unnamed: 3_level_5	C61	nan	nan	C62	C63	nan	C64
Species	Unnamed: 1_level_6	Unnamed: 2_level_6	Unnamed: 3_level_6	C71	nan	nan	C72	C73	nan	C74
UniqueID	Unnamed: 1_level_7	Unnamed: 2_level_7	Unnamed: 3_level_7	bac0	bac1	bac2	bac3	bac4	bac5	bac6

all part of Class C32, while Order C46 is part of Class C33. Note that there are many more columns in the dataset, this is just the first few. To help explain this, see the rendition of the taxonomic levels in Figure 3 below.

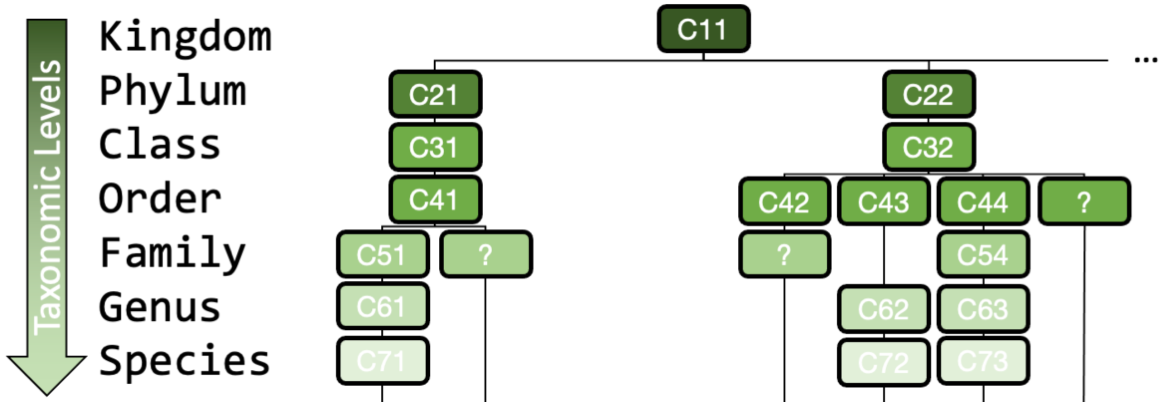


Figure 3: *Graphical illustration of the first part of the data in the multiindex columns in Figure 2. Lines indicate the connection between classes. For example, the Phylums C21 and C22 are both part of Kingdom C11. Note that this is a small portion of the overall dataframe.*

Table 1: *Description of the output labels in the DOW dataset*

Output	Full name	Units	Description
O_pH	pH	NA	Measure of how acidic or basic the basin is pH in range (7, 14] means basic pH in range [0, 7) means acidic pH = 7 means neutral
O_dO2	Dissolved Oxygen	mg/L	Amount of dissolved oxygen present in the water
O_conductivity	Conductivity	mS/cm	How well the water conducts an electric current
O_TDS	Total Dissolved Solids	mg/L	Combined amount of organic and inorganic substances dissolved in the water (Note: if the solid is present but not dissolved it is not accounted for in this quantity)
O_salinity	Salinity	Unknown	Concentration of salts in the water
O_T	Temperature	Celsius	Temperature in degrees Celsius
O_proteins	Protein content	mg/L	Amount of proteins present in the water
O_MLSS	Mixed Liquor Suspended Solids	Unknown	Total amount of solids suspended in the liquid – i.e. what’s left if you filter the water
O_MLVSS	Mixed Liquor Volatile Suspended Solids	Unknown	Total amount of organic solids suspended in the water – i.e. what gets burned off if you filter the water then burn the solids in a furnace
O_MLSSratio	MLVSS/MLSS	Unitless	Tatio of MLVSS to MLSS

Code

To get you started on the right foot, we have provided you with a starter notebook, located in **C-HACK 2022 EVENT/Projects/Provided_Code**. This notebook will walk you through the process of mounting your google drive in the notebook and loading the data. We have also provided you with a python module called `CHACKutils.py`, located in the same directory as the starter notebook. Copy these files to your teams google drive folder. In your notebook, you can import this module by following the directions in the notebook to set up your google drive and executing `import CHACKutils`. A general guide on how to do these things can be found in the following videos (Video 1, Video 2).

After you have imported the module with the above command, there are two functions which will be helpful to you:

`CHACKutils.dataloader(multiIndex=True)`
Purpose: Loads the DOW data.

- Inputs: You should pass the optional keyword argument `multiIndex=True`, which specifies that the abundance dataframe should be returned as a multi index dataframe.
- Outputs: Two dataframes. The first dataframe is the DOW bacterial abundance data, and the second is the metals and output measurements data (Table 1).

Note: Make sure you have copied the 2 csv files from the **DOW_data** folder to your team's folder. The data files must be in the same folder as your notebook and the `CHACKutils.py` file for the data loading to work correctly.

Example use of data loading function, execute the following command:

```
species_data, outputs_and_metals = CHACKutils.dataloader(
      multiIndex=True)
```

`species_data` is the dataframe of abundance data, and `outputs_and_metals` is the extra measurements dataframe. We pass `multiIndex=True` so that the `species_data` is multi indexed, meaning it has the class levels from kingdom down to species specified. See Figures 2 and 3. If you do not pass this keyword argument, the dataframe will have columns at the species level only.

`CHACKutils.getTaxonomicLevelData('TaxLevel', species_data)`
Purpose: Executing this function on your multi indexed abundance dataframe and specifying a taxonomical level, eg. "Phylum" will aggregate the dataframe to that level and return the data. View the docstring of this function to see the available taxonomic levels.

- Inputs: The first argument is a string of the taxonomic level you want to aggregate the data to, eg ‘Phylum’. The second argument is your multi indexed dataframe of abundance data.
- Outputs: A dataframe of abundance data aggregated to have columns at specific taxonomic level. Note that unclassified bacteria at the desired level will be labeled with their last known classification. For example, an unknown phylum under kingdom C12 will be labeled “C12-Unclassified”.

An example of its use is shown below:

```
phylum_data = CHACKutils.get_taxonomic_level_data(“Phylum”,
      species_data)
```

`phylum_data` is the same data as your multi indexed dataframe, but all levels lower than phylum have been summed to give you the abundance data for all phylum classes. The columns in the dataframe are the phylum.

Terminology

- *Taxonomy* - Classification of organisms. All organisms fall within a set of taxonomic classes in a hierarchy of levels. For example, one taxonomic level is “kingdom” and all animals from us humans down to fleas are part of the “animalia” kingdom. Other kingdoms include “plantae” (plants). Other taxonomic levels include genus and species. For example, brown bears (species, latin name “arctos”) are part of the “Ursus” genus, still part of the animalia kingdom.

- *Performance* - This term describes how well the basin is doing what it is supposed to do. We know that it is meant to use bacterial biomass to convert toxic biproducts such as formaldehydes into something that can be discharged safely. A metric of performance is a number that describes on a scale from “bad” to “good” how well that task is being accomplished. Note that different metrics may be used to define performance.

- *Bacterial abundance* - This is a measure of portion (think of a pie chart) of the amount of bacteria in the sample, i.e. how much of it is a specific bacteria out of 1.0. For example, if one quarter of the bacterial mass in the sample is a specific species, that species has an abundance of 0.25. Note that this does provide a measure of how much total bacteria is in the sample.

Suggestions / References for problem 2

- Gather with your team and develop a plan! Not just a technical plan, but a working plan: how will you work together, who will work on what, how will you communicate? (text, zoom, slack ?). You must submit one working final notebook, however you may find it easier to work on separate notebooks for different portions of the project, keep eachother updated, and create the final notebook as you go. Up to you!
- As the taxonomic level becomes more specific, the amount of data for any given class of bacteria is reduced. It may be worth deciding what level is specific enough to allow you to observe patterns and trends, but not so specific that your observations are not statistically sound. Consider the number of samples in each bacterial class for the entire dataset.
- Bacteria interact with each other in their ecosystem, making a species abundance dependent on the abundance of other species. Clustering algorithms may help identifying patterns and trends.
    - https://machinelearningmastery.com/clustering-algorithms-with-python/

GOOD LUCK!