****Instruction for Installing Python and Packages from requirements.txt****

****1. Installing Python****

1. Go to the official Python website: [python.org](https://www.python.org/).
2. Download the latest version of Python for your operating system (Windows, macOS, Linux).
3. Install Python. During installation, make sure to check the box ****"Add Python to PATH"**** (to add Python to the environment variables).

****2. Installing Packages from requirements.txt****

1. Open the command line (terminal) and navigate to the directory containing the requirements.txt file.
2. Run the following command:

pip install -r requirements.txt

This command will install all the necessary packages listed in the requirements.txt file.

****Instruction for Running the Example example\_5db.py****

****1. Extracting the api.zip Archive****

1. Extract the api.zip archive into a directory of your choice. The archive should contain files such as example\_5db.py, fivedb\_credentials.txt, and other necessary files.

****2. Configuring the fivedb\_credentials.txt File****

1. Open the fivedb\_credentials.txt file in a text editor.
2. Enter your login and password in the following format:

your\_login your\_password

1. Save and close the file.

****3. Running the example\_5db.py Script****

1. Open the command line (terminal) and navigate to the directory containing the example\_5db.py file.
2. Run the following command:

python example\_5db\_request\_to\_csv.py

The script will start executing, and if everything is configured correctly, you will see the query results and a graph.

****Using Saved Dictionaries****

The saved dictionaries (e.g., 5db\_process\_types.txt, 5db\_parameters.txt, 5db\_cell\_types.txt, etc.) are used to familiarize yourself with the available values in the FIVEDB database and to select appropriate elements for forming database queries. These dictionaries contain all possible options for each data type (e.g., process types, parameters, cell types, etc.) that can be used in queries.

****1. Saving Dictionaries****

The functions fiveDBManager.save\_\* save lists of all available values for each data type into text files. For example:

* 5db\_process\_types.txt contains all process types (e.g., "Migration", "Proliferation", "Apoptosis", etc.).
* 5db\_parameters.txt contains all parameters (e.g., "Emax", "EC50", "IC50", etc.).
* 5db\_cell\_types.txt contains all cell types (e.g., "T-cells", "Cancer cells", "Neurons", etc.).
* And so on for other data types.

****2. Using Dictionaries****

1. The user opens the saved files and reviews the available values. For example, they might see that the database includes the parameter "Emax" or the process type "Migration".
2. Based on this data, the user forms lists for the database query, specifying only the values they are interested in.

****3. Forming a Query****

The user passes the selected values to the fiveDBManager.select\_\* functions to set filters for the database query. For example:

* user\_process\_types = fiveDBManager.select\_process\_types(['Migration']) — selects only data related to the "Migration" process.
* user\_parameters = fiveDBManager.select\_parameters(['Emax']) — selects only data related to the "Emax" parameter.
* If the list is empty (e.g., user\_cell\_types = fiveDBManager.select\_cell\_types([])), it means no filter is applied for that data type, and all possible values will be selected.

****Example Usage****

Suppose a user wants to retrieve data for the parameter "Emax" related to the "Migration" process in experiments with "T-cells". They would do the following:

1. Open the 5db\_process\_types.txt file and see that the "Migration" process is available.
2. Open the 5db\_parameters.txt file and see that the "Emax" parameter is available.
3. Open the 5db\_cell\_types.txt file and see that the "T-cells" cell type is available.
4. Form the query:

user\_process\_types = fiveDBManager.select\_process\_types(['Migration'])

user\_parameters = fiveDBManager.select\_parameters(['Emax'])

user\_cell\_types = fiveDBManager.select\_cell\_types(['T-cells'])

1. Pass these values to the fiveDBManager.query\_data function to retrieve data matching the specified criteria.

****Advantages of Using Dictionaries****

* ****Convenience****: The user can review available values in advance and select only those they need.
* ****Flexibility****: Different filters can be combined (e.g., selecting data for a specific cell type and parameter).
* ****Error Minimization****: The user can see which values are valid, avoiding errors related to entering non-existent data.

****Explanation of Parameter Selection in the Script****

In the example\_5db.py script, there are several lines where parameters for the database query are selected. Here’s how they work:

* ****Process Types****:

user\_process\_types = fiveDBManager.select\_process\_types(['Migration'])

This selects the process types of interest. In this case, the "Migration" process is selected. You can replace 'Migration' with another process type as needed.

* ****Parameters****:

user\_parameters = fiveDBManager.select\_parameters(['Emax'])

This selects the parameters you want to retrieve. Here, the "Emax" parameter is selected. You can replace 'Emax' with another parameter as needed.

* ****Cell Types****:

user\_cell\_types = fiveDBManager.select\_cell\_types([])

This selects cell types. If the list is empty (as in the example), all cell types will be selected. You can specify specific cell types, e.g., ['T-cells', 'B-cells'].

* ****Stimulants****:

user\_stimulateds = fiveDBManager.select\_stimulated([])

This selects stimulants. If the list is empty, all stimulants will be selected. You can specify specific stimulants, e.g., ['IL-2', 'IFN-gamma'].

* ****Patient States****:

user\_patient\_states = fiveDBManager.select\_patient\_states([])

This selects patient states. If the list is empty, all states will be selected. You can specify specific states, e.g., ['Healthy', 'Disease'].

* ****Products****:

user\_products = fiveDBManager.select\_products([])

This selects products. If the list is empty, all products will be selected. You can specify specific products, e.g., ['Cytokine A', 'Cytokine B'].

* ****Daughter Cells****:

user\_daughter\_cells = fiveDBManager.select\_daughter\_cells([])

This selects daughter cells. If the list is empty, all daughter cells will be selected. You can specify specific daughter cells, e.g., ['Cell Type A', 'Cell Type B'].

* ****Regulators****:

user\_regulators = fiveDBManager.select\_regulators([])

This selects regulators. If the list is empty, all regulators will be selected. You can specify specific regulators, e.g., ['Regulator X', 'Regulator Y'].

* ****Modifiers****:

user\_modifiers = fiveDBManager.select\_modifiers([])

This selects modifiers. If the list is empty, all modifiers will be selected. You can specify specific modifiers, e.g., ['Modifier 1', 'Modifier 2'].

****Conclusion****

After configuring the parameters and running the script, you will receive data from the database and a graph visualizing this data. You can modify the query parameters to retrieve the exact data you need.