Medi_Vizz User Guide For Linux

(Developed based on the Ubuntu 23.10)

If you struggle with any part of the user guide or running the application, please contact Ms Uscinnia Dyn'ko at this address: uscinnia.dynko@ut.ee

Medi_Vizz is an application for the concurrent visualisation of medical data. It can visualise diagnoses encoded by ICD-10 classification and numerical laboratory measurements.
Medi_Vizz is not designed for data protection. Please don't leave personal identification in the names of the files or the data. It runs without access to the Internet, so there is no possibility of a data leak. However, the application's creator is not responsible for who sees your computer screen with the data while you run the application.

Installing Python to run the Medi_Vizz

Medi_Vizz is coded in Python, and to start the application, you need to have Python version 3.12 or later versions.

1. Check if Python is already installed and make the system update (Fig. 1):
- Find Terminal Application in the search bar.
- Run the command **python3 --version** to determine if Python is installed.
- Run the command **sudo apt update**
- You will be asked to enter the computer password (it won't be visible when you type).

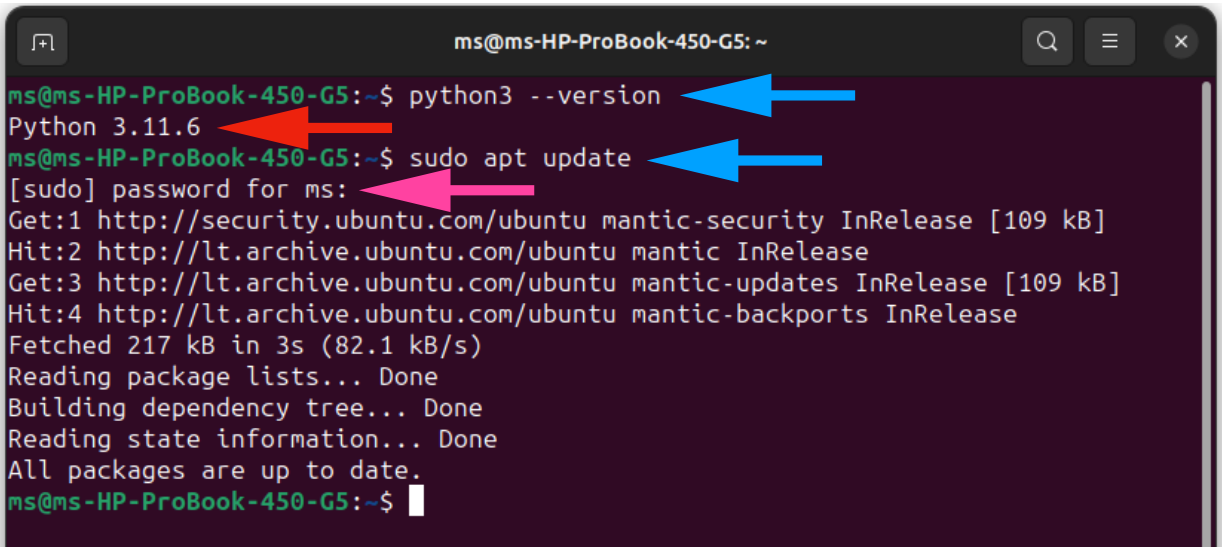

Fig. 1. Terminal output. The blue arrow points at the entered command, the red arrow points at the output after running the command, and the pink arrow points at the place where the computer password should be entered.

2. Install the Python 3.12 version and check the version after installation (Fig. 2):
- Run the command **sudo apt install python3.12**
- You will be asked to enter the computer password (it won't be visible when you type).
- Run the command **python3.12 --version**

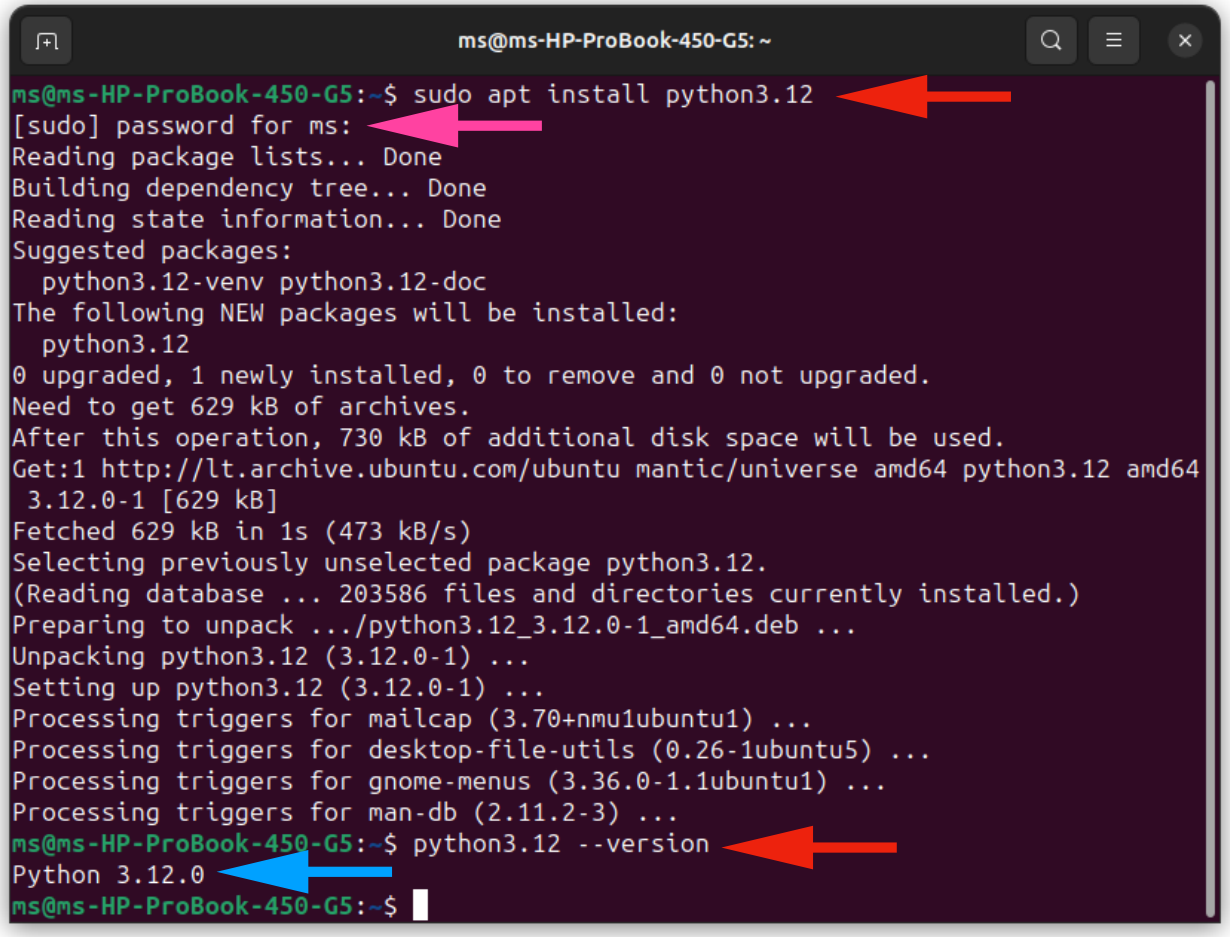

Fig. 2. Installing Python and checking the successful installation. The blue arrow points at the entered command, the red arrow points at the output after running the command, and the pink arrow points at the place where the computer password should be entered.

Installing required libraries for Medi_Vizz

1. Create new folder for the new environment where Medi_Vizz will run (Fig. 3):
- Create a new folder with the command **mkdir [name of the folder]**

e.g. **mkdir Medi_Vizz_env**
- Enter the new folder with the command **cd Medi_Vizz_env**

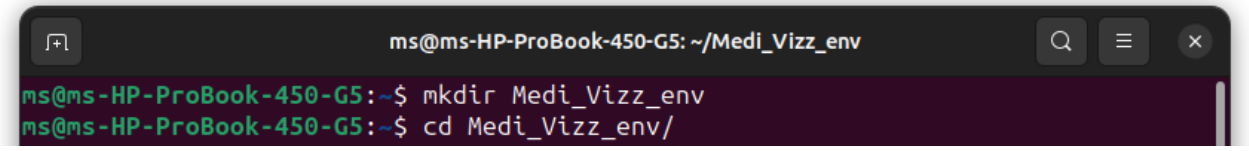

Fig. 3. Creating a folder for a future environment.

2. Create an environment where the Medi_Vizz application will run (Fig. 4):
- Make an environment with the command **python3 -m venv Medi_Vizz_env**
- Enter the bin folder of the new environment with the command

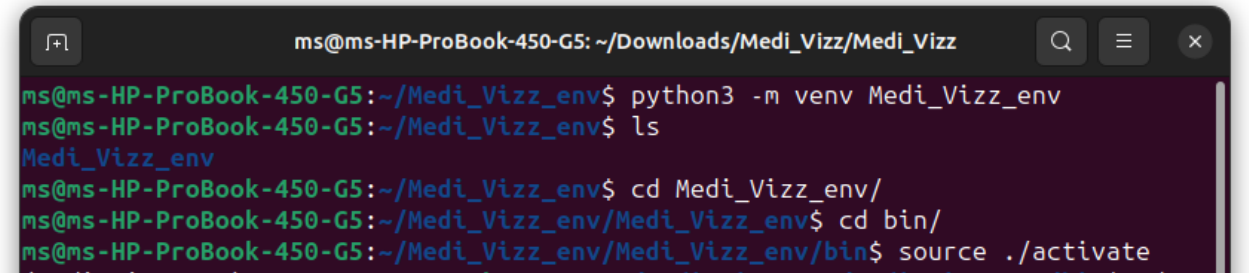

Fig. 4. Creation and activation of new environment.

cd Medi_Vizz_env/bin/
- Activate the new environment with the command **source ./activate**

When the environment is active you will see its name in the brackets highlighted by the red rectangle in Fig. 5.

3. Install the required libraries for the Medi_Vizz with the command

pip install dash plotly pandas dash-bootstrap-components python-dateutil kaleido numpy

This command installs the list of libraries required by Medi_Vizz. It will take a minute to run and gives a large output. Bits of the successful installation are shown in the Fig. 5.

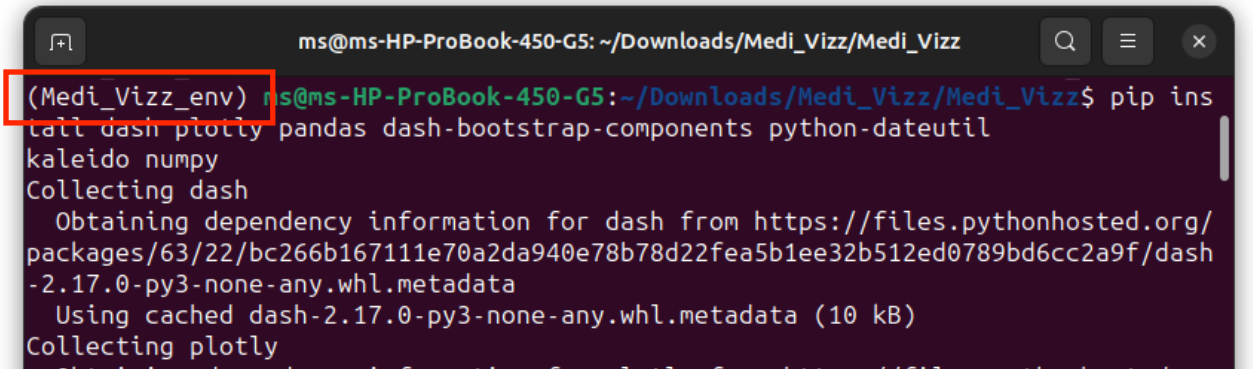

Fig. 5. Installation of the list of libraries.

Note: if at the end you see a message with a missing kaleido library (Fig. 6), run the command:
pip install kaleido

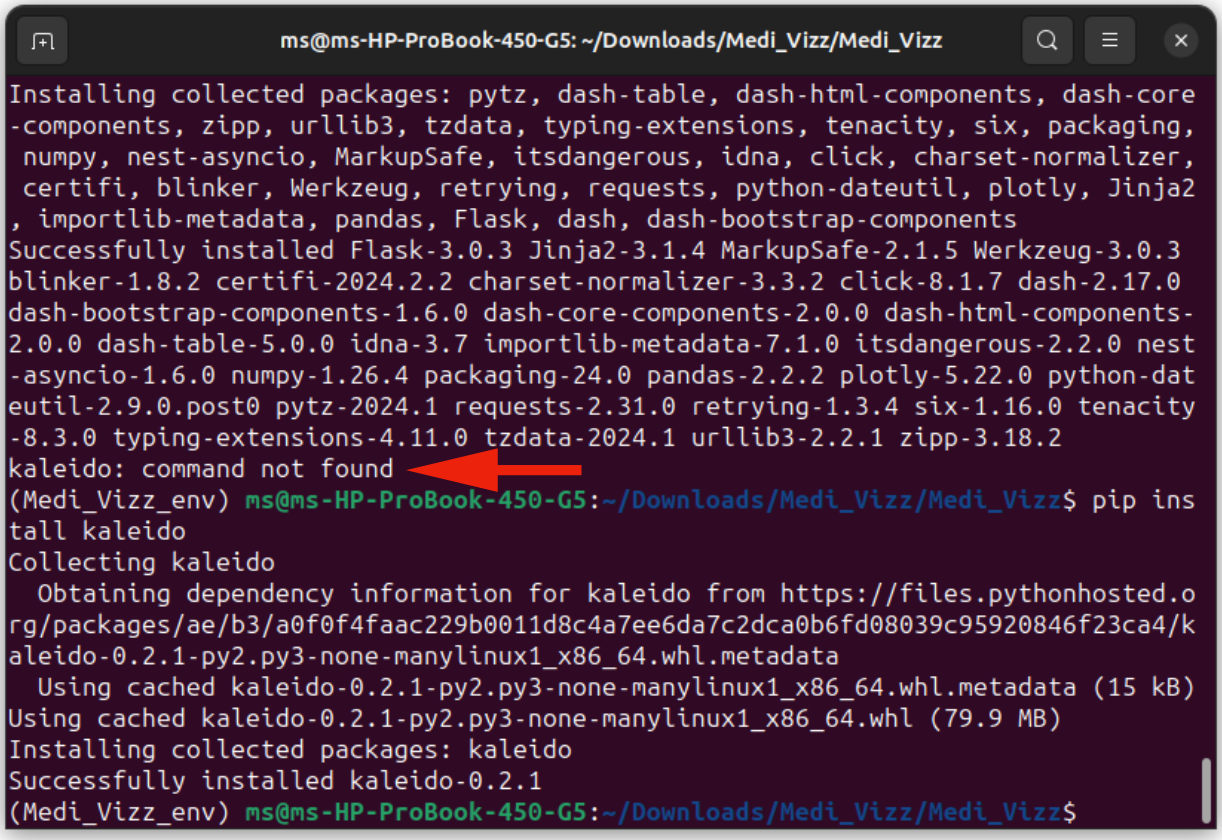

Fig. 6. Dealing with missing kaleido library. The error message is highlighted with the red arrow.

Starting the Medi_Vizz application

Download the zip folder “Medi_Vizz.zip” with the code and example files from the GitHub link https://github.com/PlanetWyh/Medi_Vizz
To start the application, you have to unpack the folder, navigate to it through the terminal, and start the application using Python.

1. In the terminal, run the command (Fig. 7):

cd ~

2. Then navigate to the downloaded Medi_Vizz folder (Fig. 7):

cd Downloads/Medi_Vizz/

3. Check that you are in the correct folder by running the command (Fig. 7):

ls

This short command displays the contents of your folder (Fig. 7).

If after running this command you see two folders, "__MACOSX" and "Medi_Vizz", run the cd **Medi_Vizz** command again (Fig. 7).

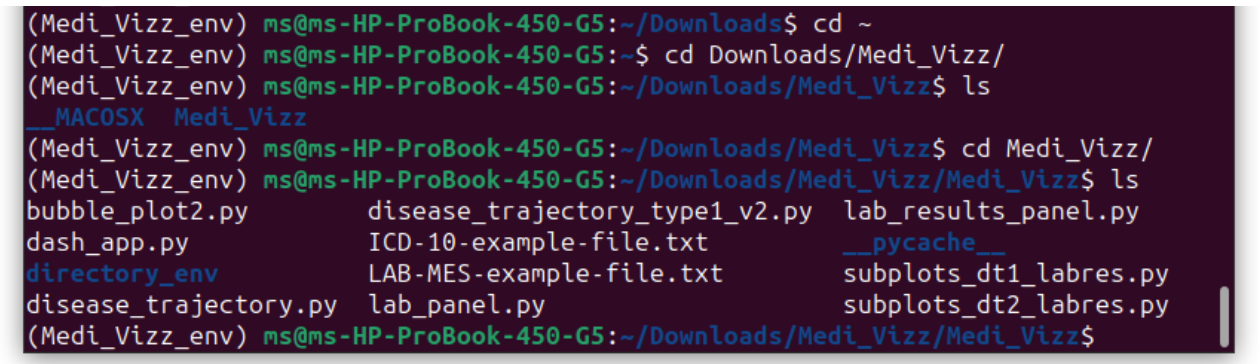

Fig. 7. Finding the Medi_Vizz folder in the terminal.

If you see bunch of .py file you are in the correct folder.

4. To start the application, run the command:

python dash_app.py

The expected output of the command is shown in the Fig. 8.

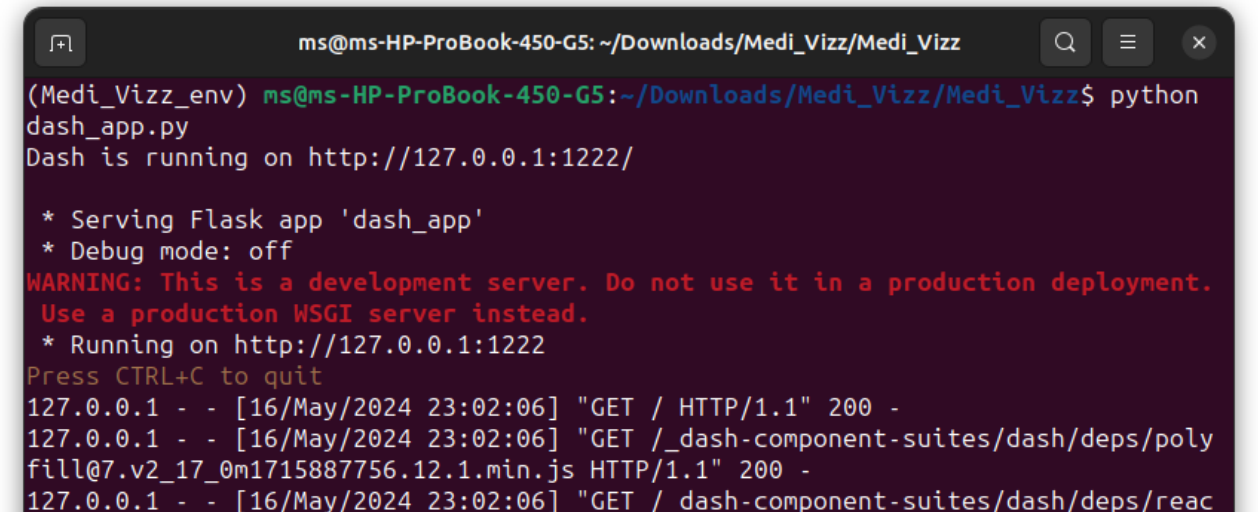

Fig. 8. Medi_Vizz application started from the macOS terminal.

4. Step 4 should open your default browser, e.g. Firefox Web Browser (Fig. 9).

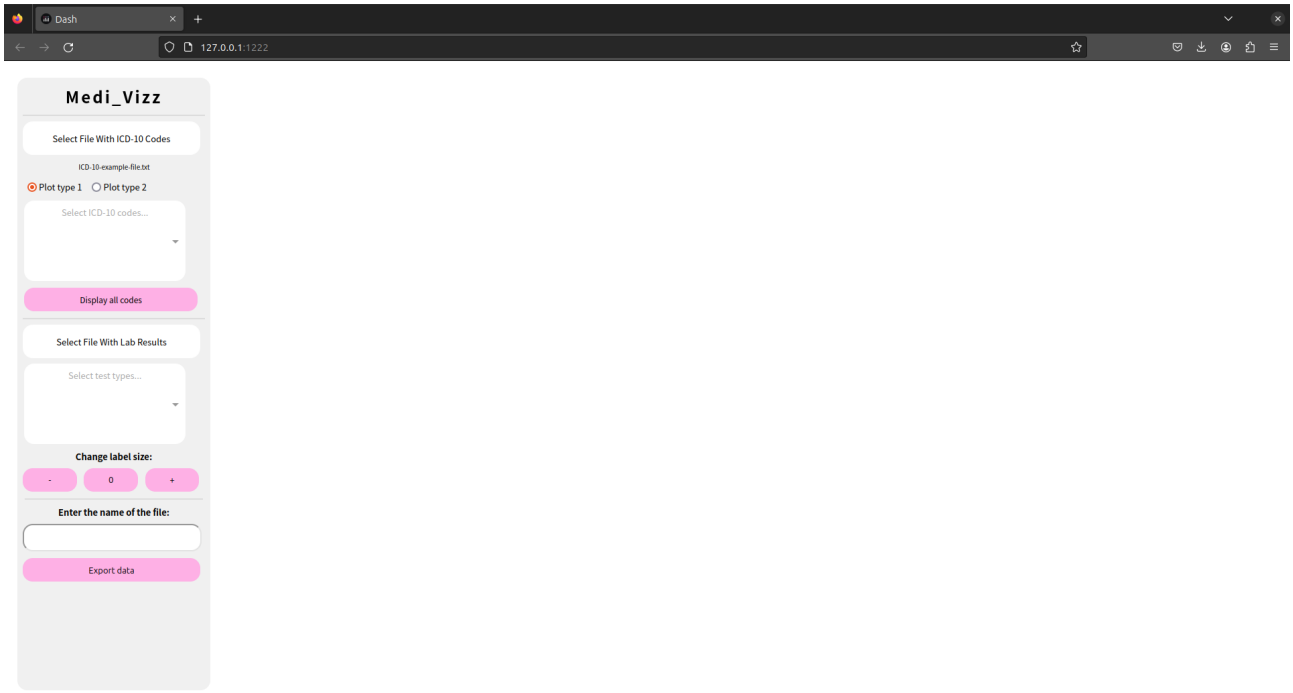

Fig. 9. Medi_Vizz application in Firefox Web Browser browser.

Data preprocessing

To use the application, you must format your data in a specific way. There are two example files in the zip folder.

ICD-10 file ("ICD-10-example-file.txt" in a zip folder)

The ICD-10 diagnoses file should include three columns:

- Numbers
- Dates: the column header should be ***Dates,*** while the column with codes should be labelled ***Codes***. Dates should be written only in the format YYYY-MM-DD without time (e.g. 1978-12-19).
- Codes. The column header should be ***Codes***. Codes should be written as in the ICD-10 classification.

Fig. 10A shows the data format example from the ICD-10-example file.txt, which you can download from GitHub.
Save the file in the **tab-delimited .txt format** (e.g. using Excel software).

ICD-10-example-file

Numbers	Dates	Codes
1	2000-11-08	R36.4
2	2001-01-09	W45
3	2001-01-09	L74.80
4	2001-01-09	S82.4
5	2002-04-20	J80.3

LAB-MES-example-file

Numbers	Dates	TestTypes	TestResultNumeric
1	2018-10-07	Test-1	0.1
2	2010-06-21	Test-1	0.5
3	2014-09-17	Test-1	0.02
4	2015-05-06	Test-1	0.9
5	2016-02-29	Test-2	6.7

Fig. 10. A) ICD-10 file example. **B)** Laboratory measurements file example.

Laboratory measurements file ("LAB-MES-example-file.txt" in a zip folder)

To make the file with laboratory results, you need to have four columns:

- Numbers
- Dates written in the format YYYY-MM-DD without time (e.g. 1978-12-19), the column labelled ***Dates***
- A column with measurement types that can include text data labelled ***TestTypes***
- A column with numeric test results labelled ***TestResultNumeric***

Fig. 10B shows the data format example from the LAB-MES-example file.txt, which you can download from GitHub.
Save the file in the **tab-delimited .txt format** (e.g. using Excel software).

Medi_Vizz functionality

Fig. 11 shows the Medi_Vizz menu and functionalities associated with each element.

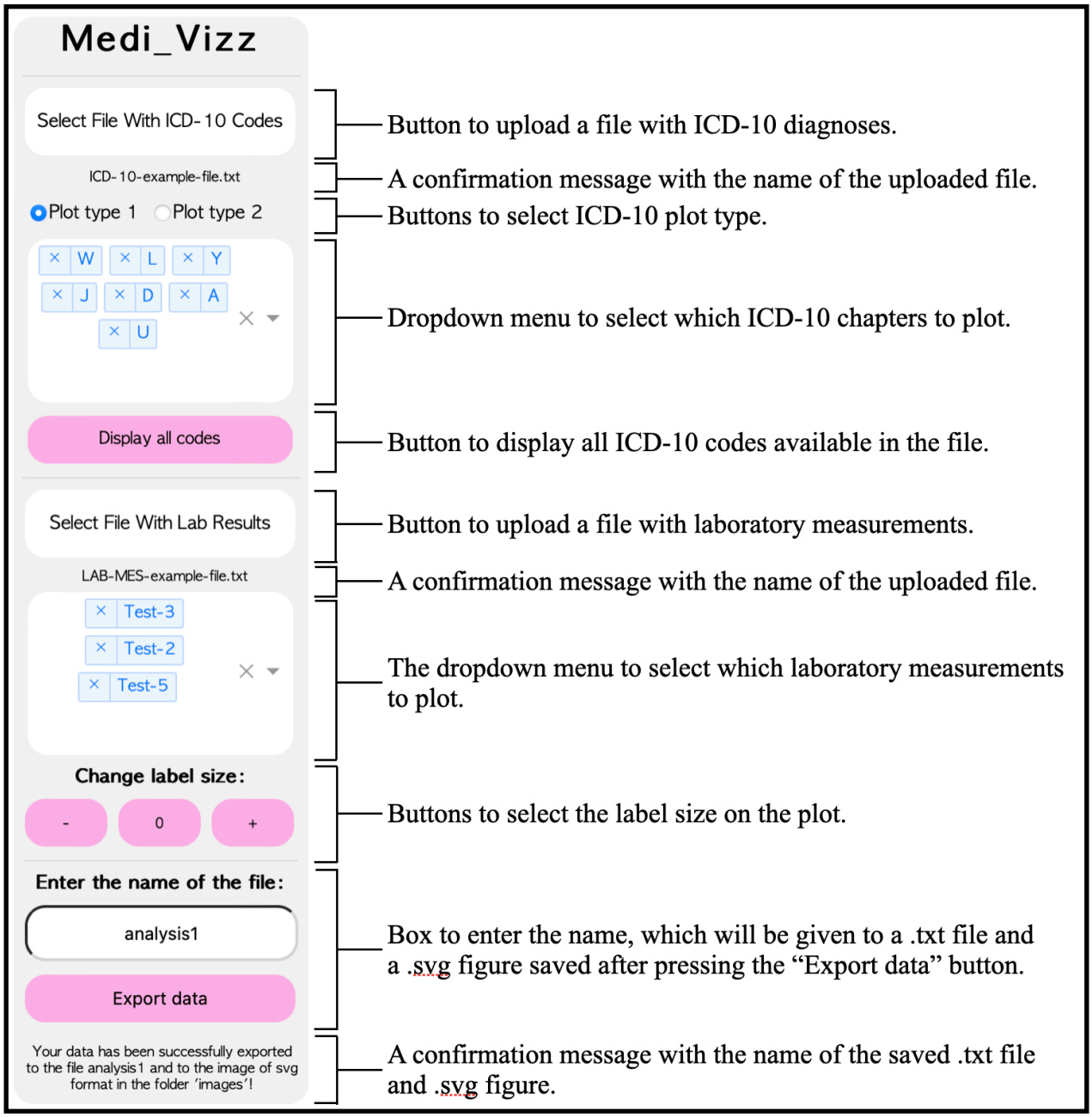

Fig. 11. Medi_Vizz menu layout with corresponding functionalities.

Medi_Vizz can generate two types of panels which only differ by how the ICD-10 diagnoses are represented. The first type of plot (Plot type 1) includes diagnoses as bubbles on the lines (Fig. 12). Line thickness shows how many diagnoses were assigned in one day, and diagnoses are displayed alphabetically. Plot type 2 shows diagnoses plotted in groups over lines (Fig. 13). The laboratory measurements panel stays the same between Plot Type 1 and Plot Type 2. Medi_Vizz considers the maximum and minimum results and distributes all the other values between them.

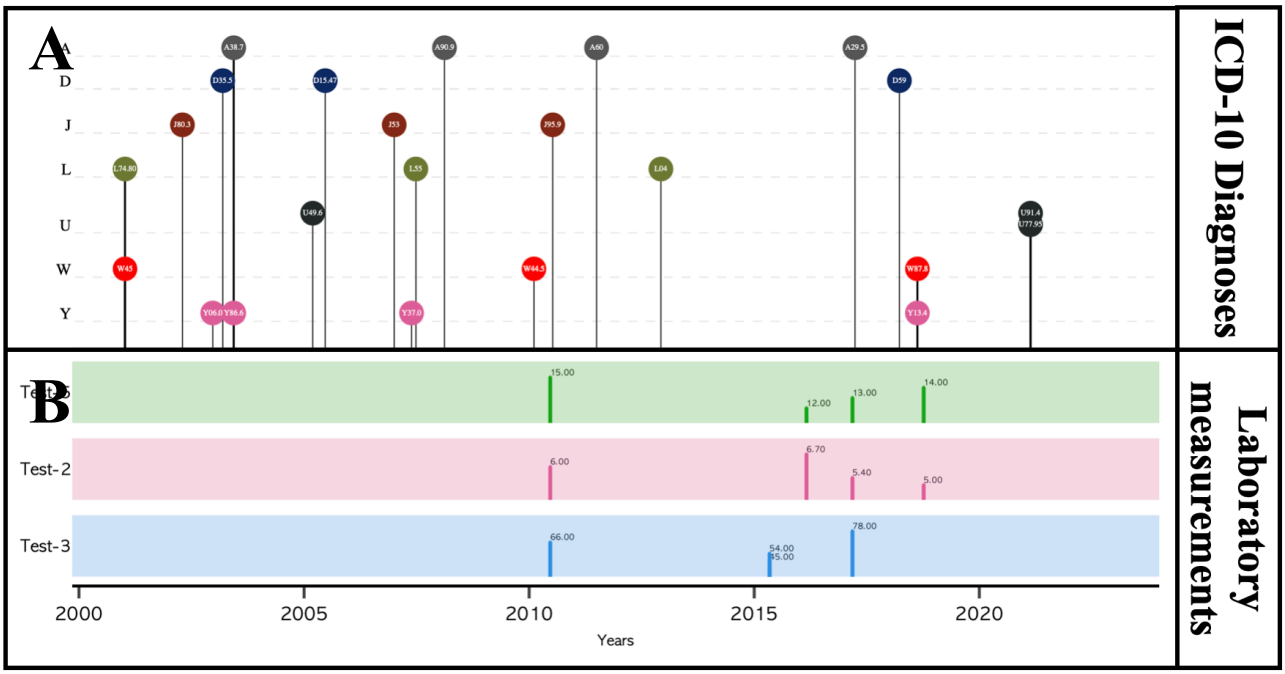

Fig. 12. Plot type 1. **A)** ICD-10 diagnoses part. **B)** Laboratory measurements part.

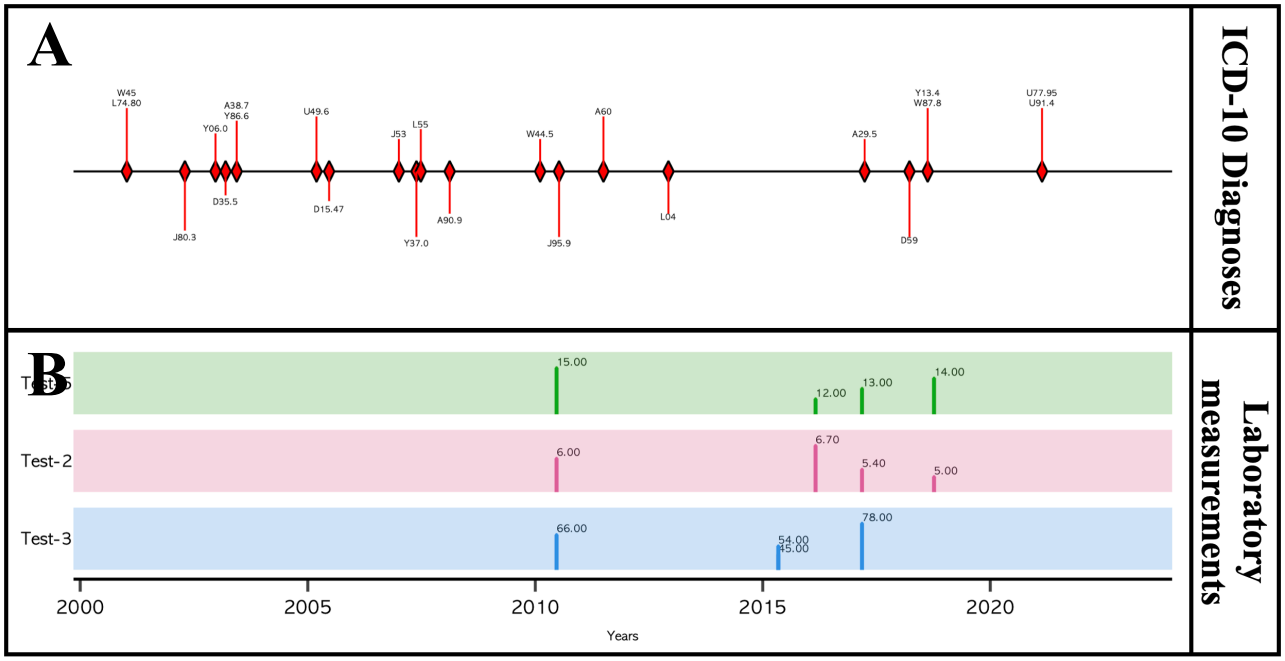

Fig. 13. Plot type 2. **A)** ICD-10 diagnoses part. **B)** Laboratory measurements part.

You can find a video example of how to use the application using this link: https://drive.google.com/file/d/1OoZ-JSpeAznnTq8mTncb2Wjbv_-UzwSE/view?usp=sharing