

Medi_Vizz User Guide

(Developed based on the macOS Sonoma Version 14.4.1)

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 3.12.2 or later versions.

1. Check if Python is already installed
 - Find Terminal Application in the search bar (Fig. 1A).
 - Run the command `python3 --version` to determine if Python is installed (Fig. 1B).
 - If Python is not installed or the version is older, follow steps 2-4.

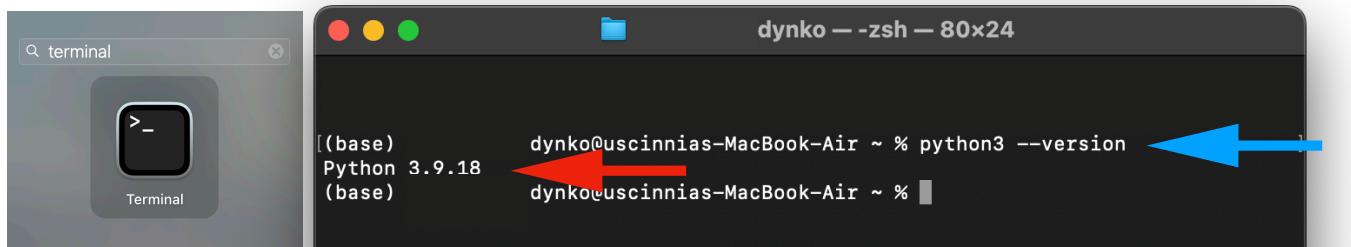


Fig. 1. A) Terminal icon in MacOS. **B)** Terminal output. The blue arrow points at the entered command, and the red arrow points at the output after running the command.

2. Go to the website <https://www.python.org/downloads/> and click on the button “Download Python 3.12.3” (Fig. 2).

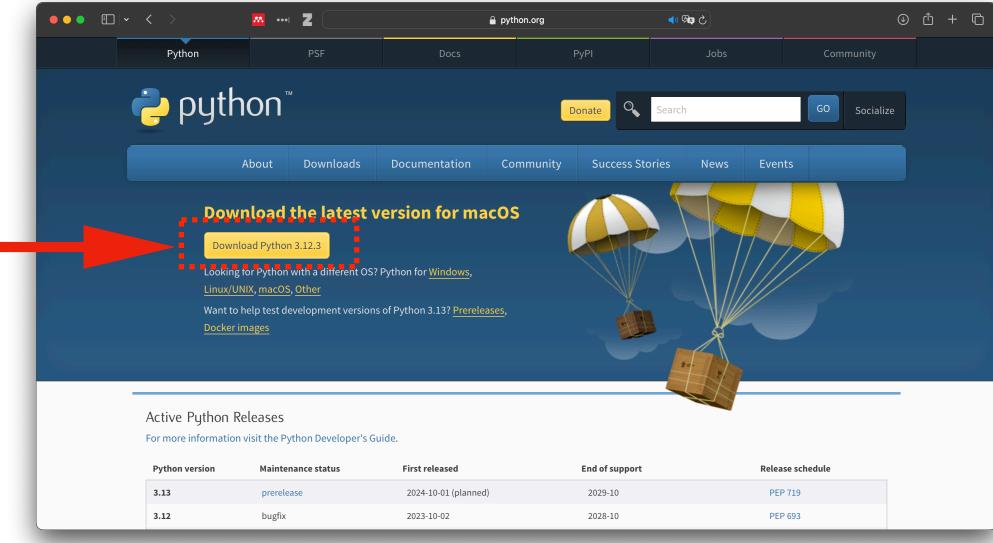


Fig. 2. Python website for downloading the installer. The red arrow points to the button that downloads Python.

3. Find the application installer in the downloads (Fig. 3).

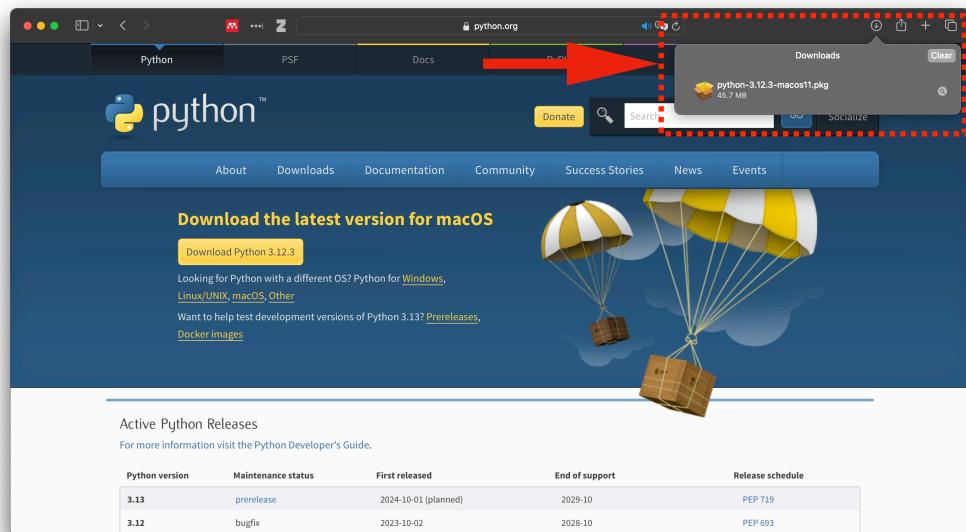
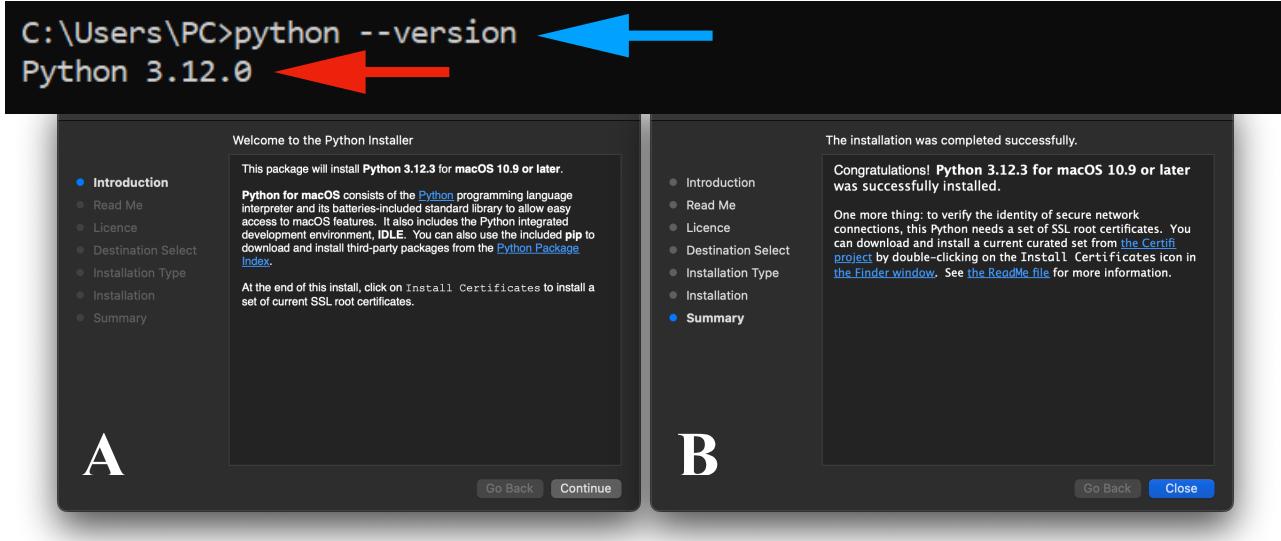


Fig. 3. Downloads in Safari browser. The red arrow points at the downloaded Python installer.



4.

Open the installer and follow the installation steps (Fig. 4A and 4B).

Fig. 4. A) Python installer first step. **4)** Python installer successful installation.

5. Verify the Python installation with the command **python3 --version** from step 1.

Installing required libraries for Medi_Vizz

You will use the pip manager to install Python libraries required by the Medi_Vizz tool.

1. Check that the pip manager is installed. Open the terminal and run the command **pip3 --version** (Fig. 5).

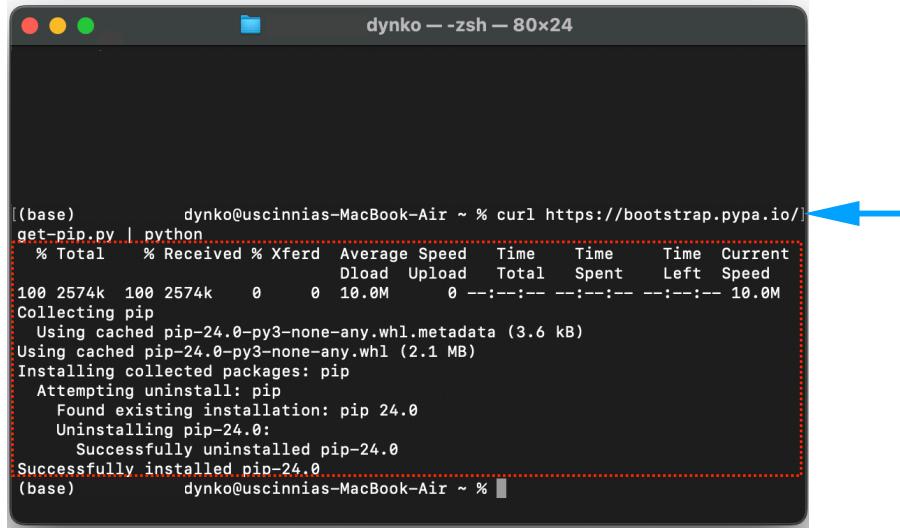
```
dynko -- zsh -- 80x24
Last login: Tue May 14 02:36:38 on ttys008
[base] dynko@uscinnias-MacBook-Air ~ % pip3 --version
pip 24.0 from /Users/dynko/opt/anaconda3/lib/python3.9/site-packages/p
(base) dynko@uscinnias-MacBook-Air ~ %
```

Fig. 5. Expected output in macOS terminal when pip manager is installed. The blue arrow indicates the entered command, while the red rectangle shows the command output.

2. If pip is not found during the previous step, run the command:

curl https://bootstrap.pypa.io/get-pip.py | python

The expected output for this command is shown in Fig. 6.



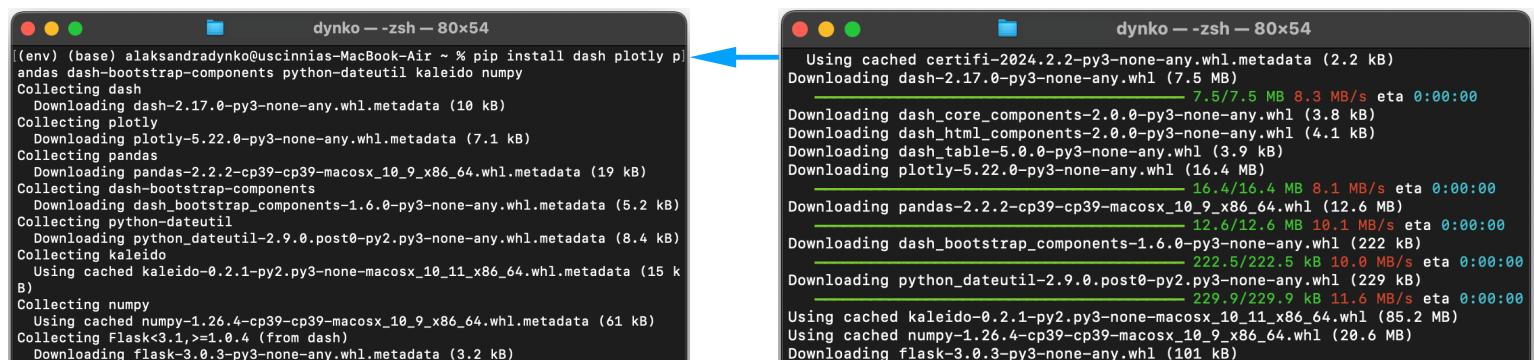
```
(base)      dynko@uscinnias-MacBook-Air ~ % curl https://bootstrap.pypa.io/get-pip.py | python
  % Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total   Spent    Left  Speed
100 2574k  100 2574k    0     0  10.0M    0 --:--:--:--:--:-- 10.0M
Collecting pip
  Using cached pip-24.0-py3-none-any.whl.metadata (3.6 kB)
Using cached pip-24.0-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 24.0
    Uninstalling pip-24.0:
      Successfully uninstalled pip-24.0
Successfully installed pip-24.0
(base)      dynko@uscinnias-MacBook-Air ~ %
```

Fig. 6. Expected output in the macOS terminal when pip manager is installed again. The blue arrow indicates the entered command, while the red rectangle shows the command output.

3. After pip is installed, in the terminal, run 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. 7.



```
(env) (base) alaksandr@uscinnias-MacBook-Air ~ % pip install dash plotly pandas dash-bootstrap-components python-dateutil kaleido numpy
Collecting dash
  Downloading dash-2.17.0-py3-none-any.whl.metadata (10 kB)
Collecting plotly
  Downloading plotly-5.22.0-py3-none-any.whl.metadata (7.1 kB)
Collecting pandas
  Downloading pandas-2.2.2-cp39-cp39-macosx_10_9_x86_64.whl.metadata (19 kB)
Collecting dash-bootstrap-components
  Downloading dash_bootstrap_components-1.6.0-py3-none-any.whl.metadata (5.2 kB)
Collecting python-dateutil
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting kaleido
  Using cached kaleido-0.2.1-py2.py3-none-macosx_10_11_x86_64.whl.metadata (15 kB)
Collecting numpy
  Using cached numpy-1.26.4-cp39-cp39-macosx_10_9_x86_64.whl.metadata (61 kB)
Collecting Flask<3.1,>=1.0.4 (from dash)
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
```

```
Using cached certifi-2024.2.2-py3-none-any.whl.metadata (2.2 kB)
Downloading dash-2.17.0-py3-none-any.whl (7.5 MB)
  7.5/7.5 MB 8.3 MB/s eta 0:00:00
Downloading dash_core_components-2.0.0-py3-none-any.whl (3.8 kB)
Downloading dash_html_components-2.0.0-py3-none-any.whl (4.1 kB)
Downloading dash_table-5.0.0-py3-none-any.whl (3.9 kB)
Downloading plotly-5.22.0-py3-none-any.whl (16.4 MB)
  16.4/16.4 MB 8.1 MB/s eta 0:00:00
Downloading pandas-2.2.2-cp39-cp39-macosx_10_9_x86_64.whl (12.6 MB)
  12.6/12.6 MB 10.1 MB/s eta 0:00:00
Downloading dash_bootstrap_components-1.6.0-py3-none-any.whl (222 kB)
  222.5/222.5 kB 10.0 MB/s eta 0:00:00
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
  229.9/229.9 kB 11.6 MB/s eta 0:00:00
Using cached kaleido-0.2.1-py2.py3-none-macosx_10_11_x86_64.whl (85.2 kB)
Using cached numpy-1.26.4-cp39-cp39-macosx_10_9_x86_64.whl (20.6 kB)
Downloading flask-3.0.3-py3-none-any.whl (101 kB)
```

Fig. 7. Partial expected output of installing list of required packages using pip. The command used is indicated by the blue 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 command line/terminal, and start the application using Python.

1. In the terminal, run the command (Fig. 8):

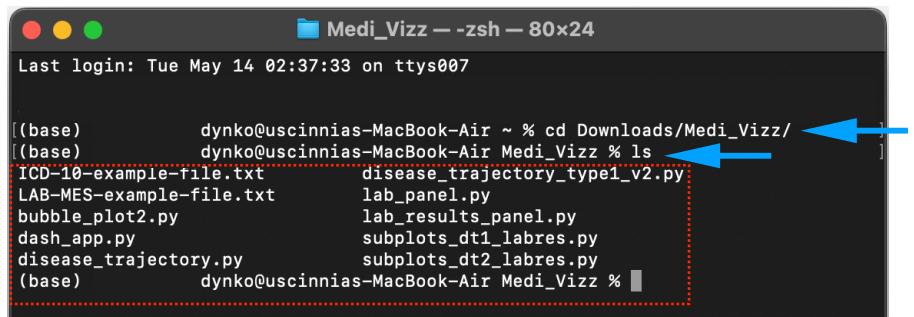
```
cd Downloads/Medi_Vizz/
```

2. Check that you are in the correct folder by running the command:

```
ls
```

This short command displays the contents of your folder (Fig. 8).

Fig. 8. Terminal in the Medi_Vizz directory. Blue arrows indicate commands entered, the red rectangle shows the output of the **ls** command — contents of the Medi_Vizz folder.



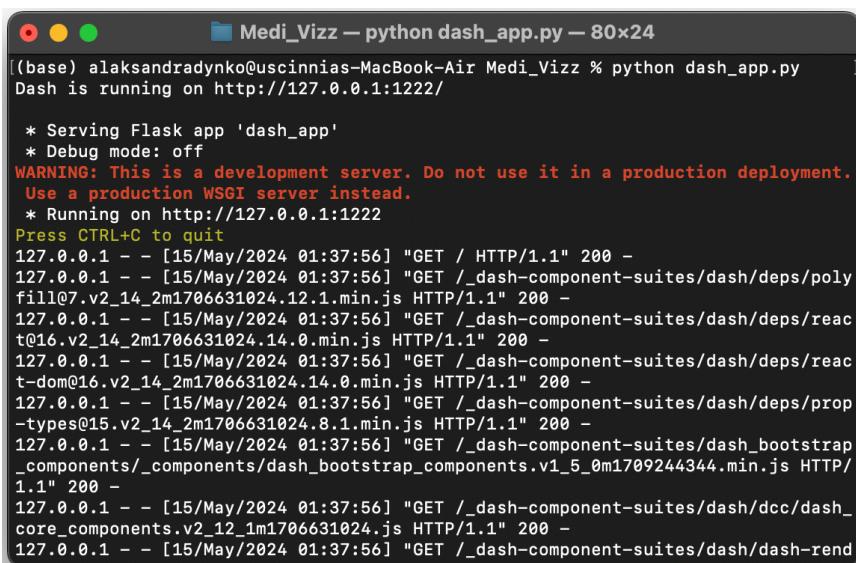
```
Last login: Tue May 14 02:37:33 on ttys007

(base)      dynko@uscinnias-MacBook-Air ~ % cd Downloads/Medi_Vizz/ ←
(base)      dynko@uscinnias-MacBook-Air Medi_Vizz % ls ←
ICD-10-example-file.txt      disease_trajectory_type1_v2.py
LAB-MES-example-file.txt    lab_panel.py
bubble_plot2.py              lab_results_panel.py
dash_app.py                 subplots_dt1_labres.py
disease_trajectory.py       subplots_dt2_labres.py
(base)      dynko@uscinnias-MacBook-Air Medi_Vizz %
```

3. To start the application, run the command:

```
python dash_app.py
```

The expected output of the command is shown in the Fig. 9.



```
Medi_Vizz — python dash_app.py — 80x24

(base) alaksandradynko@uscinnias-MacBook-Air Medi_Vizz % python dash_app.py
Dash is running on http://127.0.0.1:1222/
* Serving Flask app 'dash_app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on http://127.0.0.1:1222
Press CTRL+C to quit
127.0.0.1 - - [15/May/2024 01:37:56] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2024 01:37:56] "GET /_dash-component-suites/dash/deps/polyfill@0.2.14_2m1706631024.12.1.min.js HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2024 01:37:56] "GET /_dash-component-suites/dash/deps/react@16.2.14_2m1706631024.14.0.min.js HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2024 01:37:56] "GET /_dash-component-suites/dash/deps/react-dom@16.2.14_2m1706631024.14.0.min.js HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2024 01:37:56] "GET /_dash-component-suites/dash/deps/prop-types@15.2.14_2m1706631024.8.1.min.js HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2024 01:37:56] "GET /_dash-component-suites/dash_bootstrap_components/_components/dash_bootstrap_components.v1.5_0m1709244344.min.js HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2024 01:37:56] "GET /_dash-component-suites/dash/dcc/dash_core_components.v2.12_1m1706631024.js HTTP/1.1" 200 -
127.0.0.1 - - [15/May/2024 01:37:56] "GET /_dash-component-suites/dash/dash-rend
```

Fig. 9. Medi_Vizz application started from the macOS terminal.

4. Step 3 should open your default browser, e.g. Safari (Fig. 10).

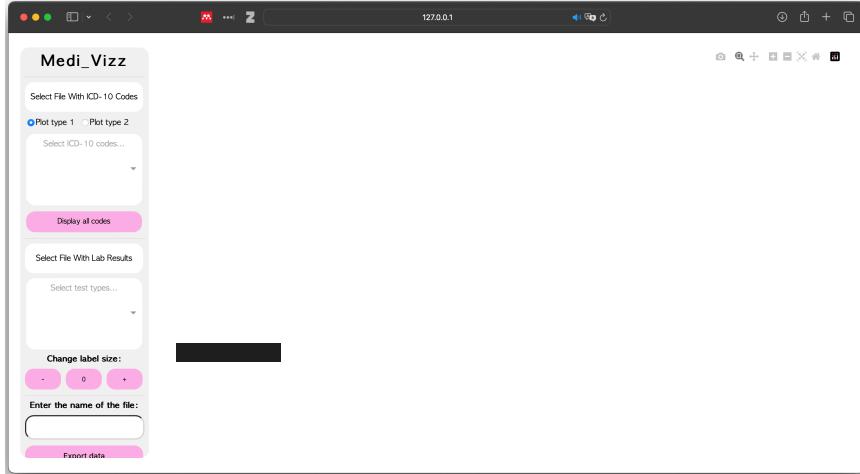


Fig. 10. Medi_Vizz application in Safari browser.

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. 11A 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			LAB-MES-example-file			
Numbers	Dates	Codes	Numbers	Dates	TestTypes	TestResultNumeric
1	2000-11-08	R36.4	1	2018-10-07	Test-1	0.1
2	2001-01-09	W45	2	2010-06-21	Test-1	0.5
3	2001-01-09	L74.80	3	2014-09-17	Test-1	0.02
4	2001-01-09	S82.4	4	2015-05-06	Test-1	0.9
5	2002-04-20	J80.3	5	2016-02-29	Test-2	6.7

Fig. 11. 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. 11B 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. 12 shows the Medi_Vizz menu and functionalities associated with each element.

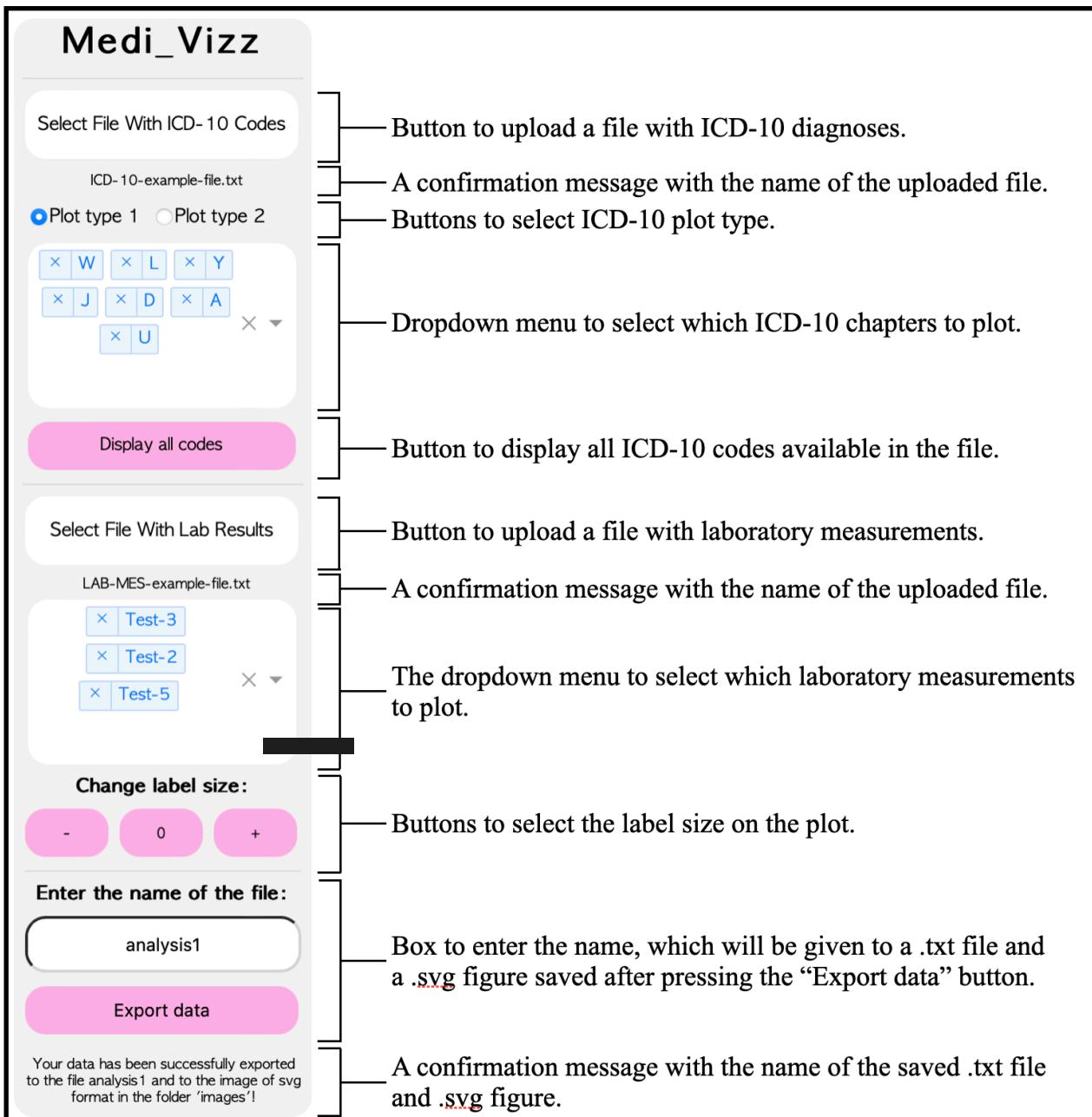


Fig. 12. Medi_Vizz menu layout with corresponding functionalities.

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. 13). 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. 14). 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.

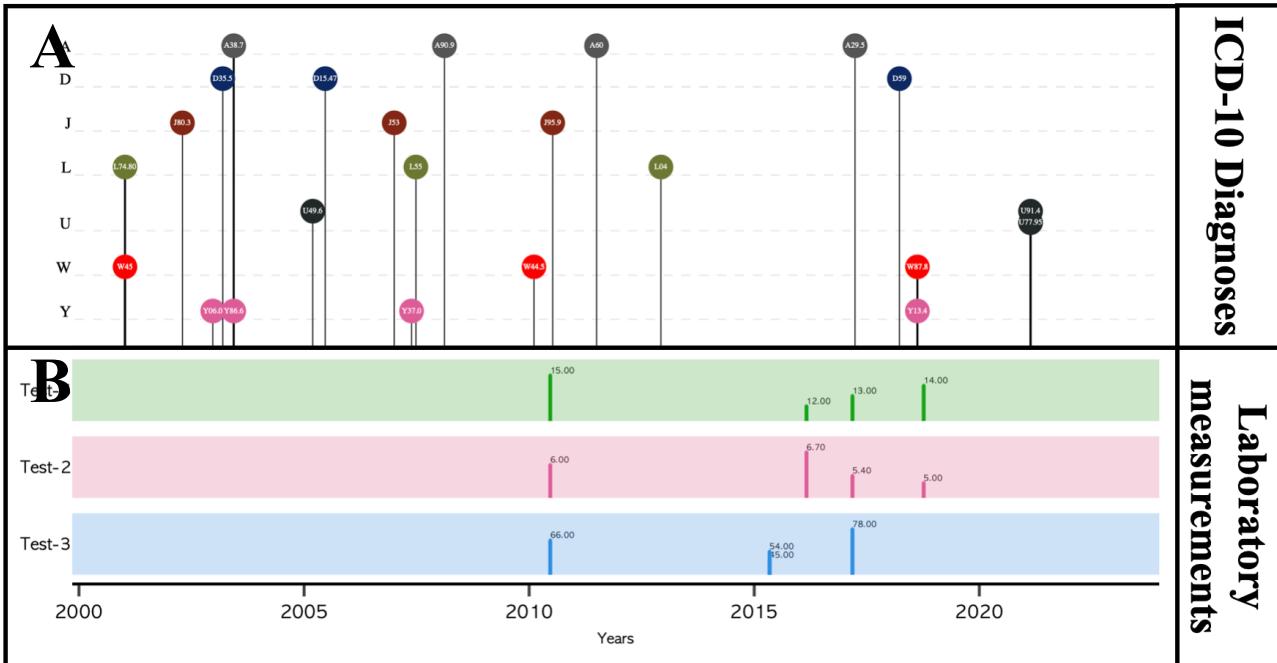


Fig. 13. Plot type 1. **A)** ICD-10 diagnoses part. **B)** Laboratory measurements part.

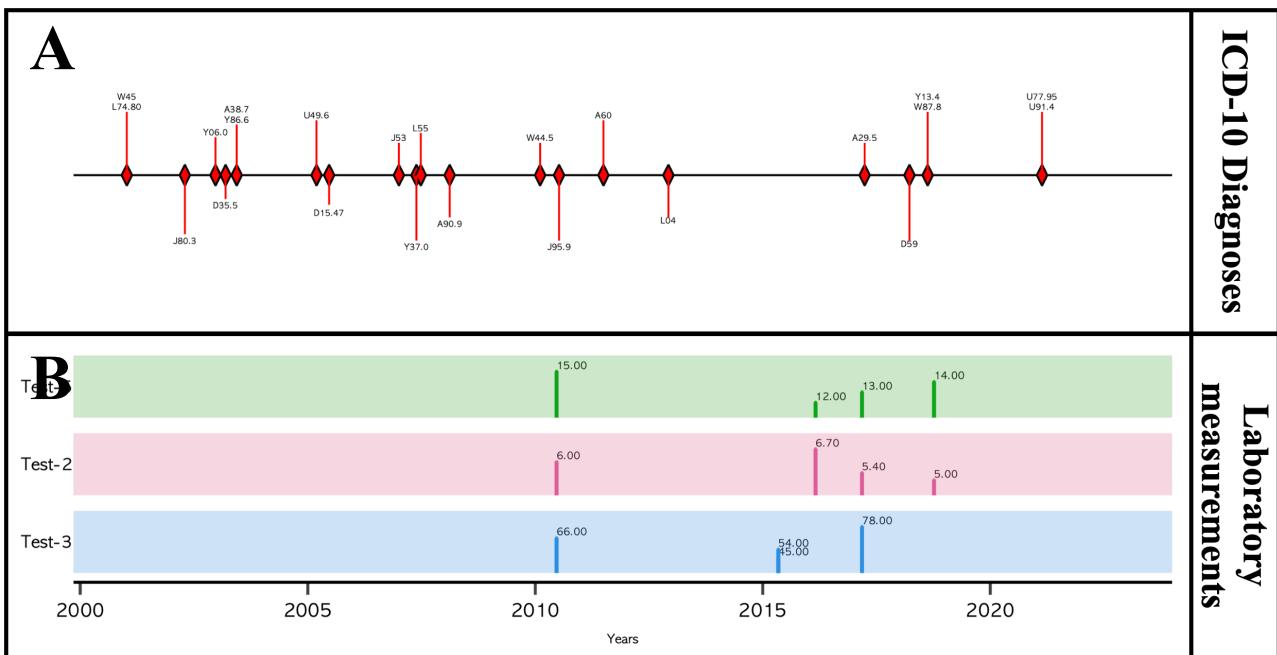


Fig. 14. Plot type 2. **A)** ICD-10 diagnoses part. **B)** Laboratory measurements part.

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