Documentation MPSA Viewer

Julien Cappèle, Claude Didierjean, Frédérique Favier

Software version alpha 0.2; Documentation version 0.1; November 2020

Contents

1 Introduction

MPSA Viewer has been created to visually emphasize multi-protein structural alignment by using both PyMOL and mTM-align, two robust and powerful analysis tools. Firstly, the mTM-align server will handle your structures using TM-align and produce a protein data bank file containing all structurally aligned atoms of your proteins, and a sequence alignment based on the later. Analyzing those results with MPSA Viewer means that all software detectable similarities and differences within your alignment will be shown visually using PyMOL.

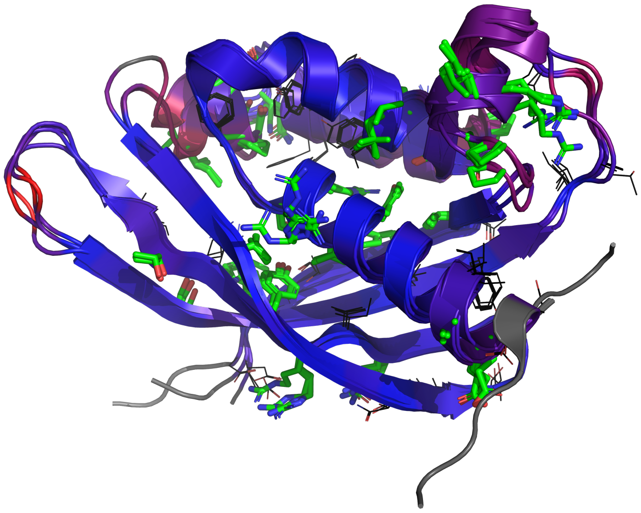


Figure 1: MPSA Viewer representation of a structural alignment in PyMOL

In the Figure 1, you can see an alignment performed by mTM-align with 4 structures (2cc3, 4kz1, 4lso, 4mei) shown in PyMOL. Green and black lateral chains represent respectively identical and same class amino acids if all structures are aligned at this exact position in space. If one of those structure would be too far, none of those amino acids would show up. Additionally, a backbone gradient coloration is applied based on the RMSD of alpha carbons of those amino acids. In this example, blue means close to 0 Å, red means close or above to 2 Å. A grey coloration is applied whenever the highest distance between two alpha carbons at a given position exceeds 4 Å, or if mTM-align placed a gap (-) in the alignment at this position for one or more structure. Those parameters are the default for MPSA Viewer, but those parameters can be changed pretty easily to your preferences.

In this documentation we will go through the installation process and the every-day use of the software. As always with third-party software, make sure you downloaded it for free from a trust worthy source. Authors of the software are usually a good start!

2 How install MPSA Viewer

2.1 Compatibility

As of now, MPSA Viewer *should* be natively compatible with any recent linux distribution, and is compatible with Windows 10 (20H2). MPSA Viewer has not been tested on OSX systems (yet).

2.2 Python

MPSA Viewer is a python script, meaning that you need to install Python. Python is *usually* natively installed on Unix (Linux & OSX) systems, however, you will need a recent version of Python. For Windows users, you will have to install it.

Python 3.8.5 is one of the latest release of Python as of now and is recommended to use this software. MPSA Viewer might be compatible with Python 3.4, but has not been tested with those versions.

Here is a complete set of URLs to the python documentation on all platforms:

Windows : https://docs.python.org/3/using/windows.html

Linux : https://docs.python.org/3/using/unix.html

Mac : https://docs.python.org/3/using/mac.html

2.3 Installation and folder preparation

MPSA Viewer is portable, meaning it is contained into a single script file that you can put anywhere you want on your system (except in your system files, obviously). On startup, MPSA Viewer will create a folder named MPSA_Projects that will be used to store all of your projects and data. If you move the script file, the project folder won't be used anymore and another one will be created. The best way to move the software is to also move the projects folder with it.

3 Some things to know about MPSA Viewer

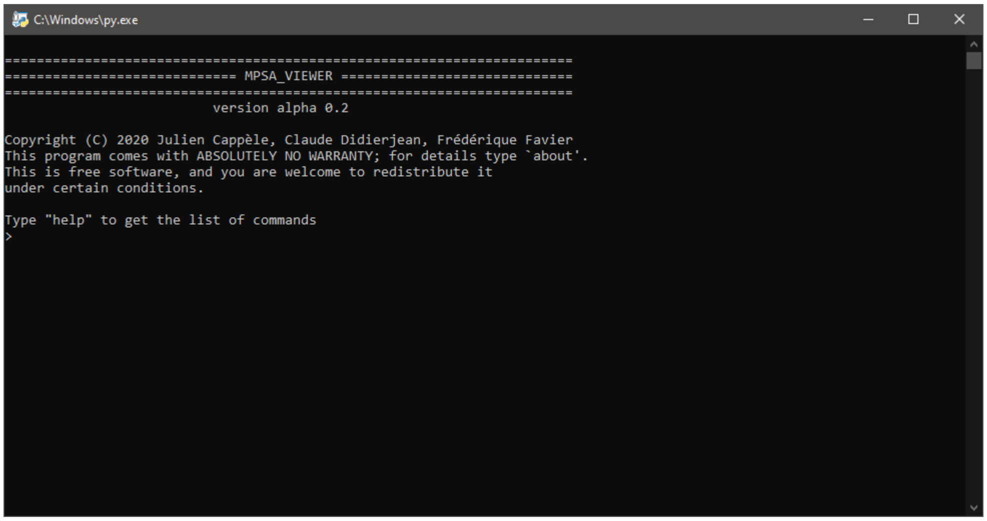
3.1 Terminal

MPSA Viewer will work as a command line terminal. Without any additional library for Python, this means that it will be very basic without auto-completion, previous command history or any advanced command line terminal extensions. But it will get the job done.

In order to start the terminal, you have to use your operating system terminal first, navigate to the executable folder and use the command :

```
$ python mpsa_executable.py
```

This will start the terminal and should look like this:



```
=======================================================================
============================= MPSA_VIEWER =============================
=======================================================================
                          version alpha 0.2

Copyright (C) 2020 Julien Cappèle, Claude Didierjean, Frédérique Favier
This program comes with ABSOLUTELY NO WARRANTY; for details type `about'.
This is free software, and you are welcome to redistribute it
under certain conditions.

Type "help" to get the list of commands
>
```

Figure 2: Default look of MPSA Viewer in Windows terminal

Starting now, all commands that are typed within the software terminal will start with the superior sign (>), indicating a command in MPSA Viewer.

3.2 Commands

3.2.1 help

When the program starts, you will be prompted for a command. If you don't remember the syntax of a command, you can ask the program for help. By using help, the terminal will print a basic description of every implemented commands. Example of use:

```
> help
```

3.2.2 new_project <name> (<template>)

The command new_project creates a folder within the MPSA_Project folder alongside the script executable file. You have to specify the name of your project and the folder will be given the same name. Inside your new project folder, will be created everything needed for MPSA Viewer to work, especially some configuration files that are important.

Make sure to check the configuration file and specify your PyMOL path, and specify your desired structure names in the structure configuration file for every new project.

Example of use:

```
> new_project gfp_18_Oct_2020
```

If you want to create a project based on another one, there is an optional argument <template> that you can use to specify the project that you want to use as a template for the new one. Your preferences will be imported from the template to your new project.

Example of use:

```
> new_project gfp_21_Oct_2020 gfp_18_Oct_2020
```

3.2.3 zip <name>

The command zip will prepare your desired structure files that you have specified, will download them if possible and ask for you to place your own structures in the correct folder. Once all structures are present, they will be searched for seleno-methionines that are unfortunately not compatible with mTM-align for now. Finally, all corrected structures are compressed into a zip file, ready to be sent on the mTM-align server to be worked on.

Example of use:

```
> zip gfp_21_Oct_2020
```

3.2.4 fetch <name> <url>

The command fetch will, as its name suggest, fetch results from mTM-align server based on the url of the finished job and place them in the correct folder within your project.

Example of use:

```
> fetch gfp_21_Oct_2020 https://yanglab.nankai.edu.cn/mTM-align/output/...
```

3.2.5 analysis <name>

The command analysis is the main command of MPSA Viewer and will need mTM-align results and the structures. If you used the commands zip and fetch, you should be set already. If not, you have to place all needed structures in the "Processed_Structure" folder, and the mTM-align results in "MSTA_result". This command will both use PyMOL and the mTM-align result to produce a PyMOL session file.

Example of use:

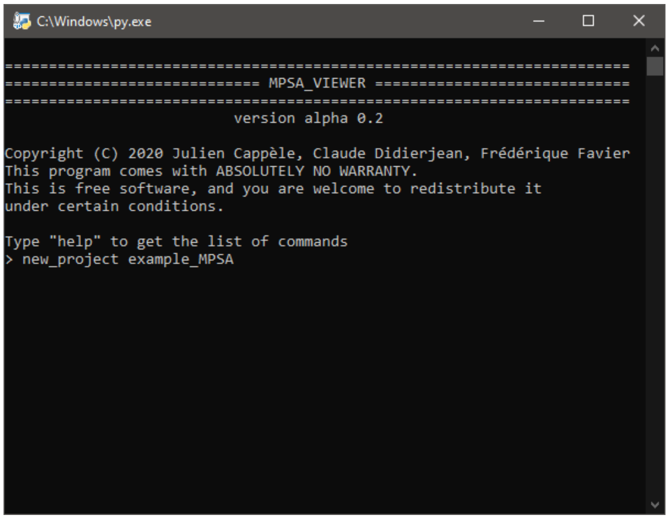
```
> analysis gfp_21_Oct_2020
```

3.3 Example of a project with MPSA Viewer

Let's dive together into the software and try it with some structures. Because a multiple protein alignment can only be used with very similar structures, we choose a set of 4 similar structures for this example.

Those are 2cc3, 4kz1, 4lso and 4mei. These 4 proteins are known as VirB8-like proteins and are involved in gram negative bacteria conjugation.

Starting with a new project, I will give it the name of "example_MPSA"

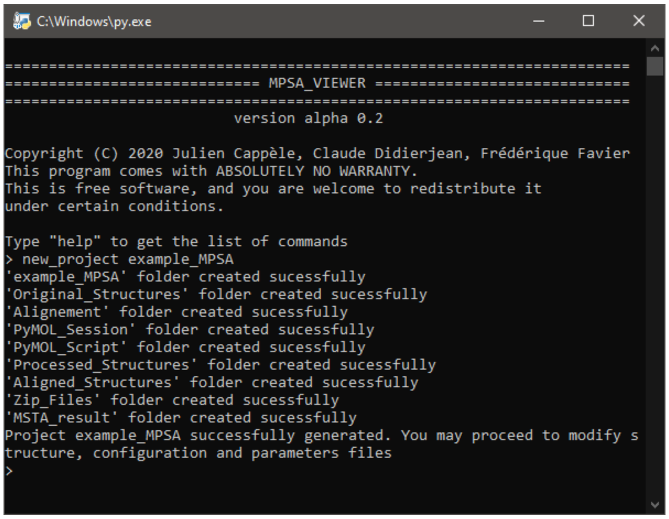


```
=======================================================================
============================= MPSA_VIEWER =============================
=======================================================================
                          version alpha 0.2

Copyright (C) 2020 Julien Cappèle, Claude Didierjean, Frédérique Favier
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions.

Type "help" to get the list of commands
> new_project example_MPSA
```

Figure 3: Creating a new project

The last message tells us that we need to modify configurations files. Because the software use PyMOL, the first thing that we need to do is to change that into the configuration_file.txt of the project. In my case, PyMOL is located at this path :

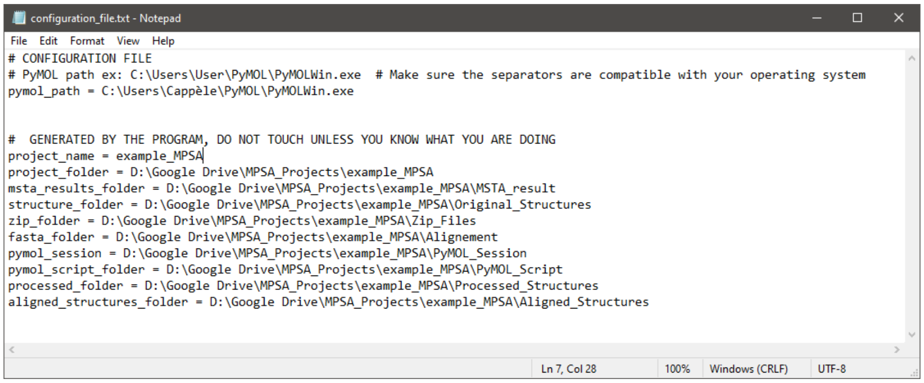
```
C:\Users\Cappèle\PyMOL\PyMOLWin.exe
```



```
=======================================================================
============================ MPSA_VIEWER ==============================
=======================================================================
                          version alpha 0.2

Copyright (C) 2020 Julien Cappèle, Claude Didierjean, Frédérique Favier
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions.

Type "help" to get the list of commands
> new_project example_MPSA
'example_MPSA' folder created sucessfully
'Original_Structures' folder created sucessfully
'Alignement' folder created sucessfully
'PyMOL_Session' folder created sucessfully
'PyMOL_Script' folder created sucessfully
'Processed_Structures' folder created sucessfully
'Aligned_Structures' folder created sucessfully
'Zip_Files' folder created sucessfully
'MSTA_result' folder created sucessfully
Project example_MPSA successfully generated. You may proceed to modify s
tructure, configuration and parameters files
>
```

Figure 4: The new project has been created

The path could be different if you are using Linux or Mac. Make sure that your path is exactly pointing to the executable of PyMOL.



```
# CONFIGURATION FILE
# PyMOL path ex: C:\Users\User\PyMOL\PyMOLWin.exe  # Make sure the separators are compatible with your operating system
pymol_path = C:\Users\Cappèle\PyMOL\PyMOLWin.exe


#  GENERATED BY THE PROGRAM, DO NOT TOUCH UNLESS YOU KNOW WHAT YOU ARE DOING
project_name = example_MPSA
project_folder = D:\Google Drive\MPSA_Projects\example_MPSA
msta_results_folder = D:\Google Drive\MPSA_Projects\example_MPSA\MSTA_result
structure_folder = D:\Google Drive\MPSA_Projects\example_MPSA\Original_Structures
zip_folder = D:\Google Drive\MPSA_Projects\example_MPSA\Zip_Files
fasta_folder = D:\Google Drive\MPSA_Projects\example_MPSA\Alignement
pymol_session = D:\Google Drive\MPSA_Projects\example_MPSA\PyMOL_Session
pymol_script_folder = D:\Google Drive\MPSA_Projects\example_MPSA\PyMOL_Script
processed_folder = D:\Google Drive\MPSA_Projects\example_MPSA\Processed_Structures
aligned_structures_folder = D:\Google Drive\MPSA_Projects\example_MPSA\Aligned_Structures
```

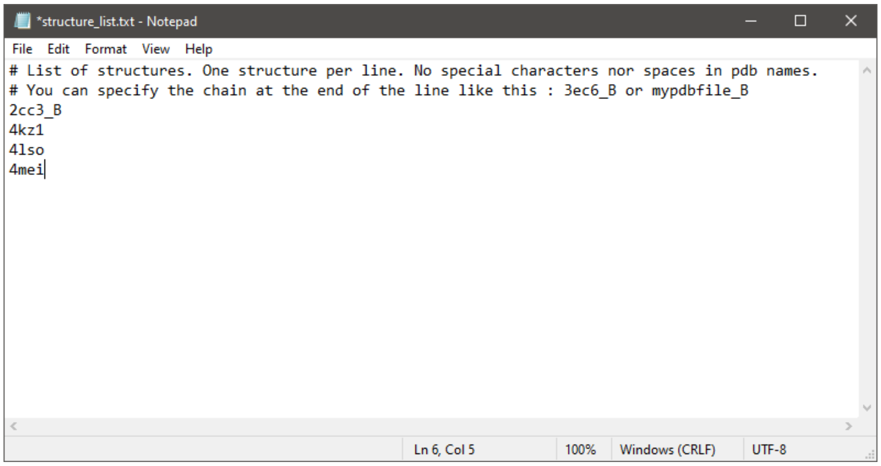
Figure 5: The configuration file

Now that PyMOL path is set, we need to specify our structures into the structure list file "structure_list.txt"

We can enter at each line a new pdb name. If you have a non published structure, just enter its name here, and you will be asked to manually place the pdb file in the "Original_Structures" folder. It is only compatible with pdb files. By default, the chain A will be used for the alignment but if you want to test another chain, you can specify it by using an underscore and the letter of the chain. This can only be used for mono-letters chains. (Figure 6)

If you want to change the PyMOL session visuals, you can modify any variable you want in the "parameters.txt" file, but we will use the default for this example, because we really like it ! ;-)

Everything is set and we can create the zip file by using the "zip" command.



```
# List of structures. One structure per line. No special characters nor spaces in pdb names.
# You can specify the chain at the end of the line like this : 3ec6_B or mypdbfile_B
2cc3_B
4kz1
4lso
4mei
```

Figure 6: Structures list

```
> zip example_MPSA
```

After a longer and quite cryptic output, all structures have been downloaded from the protein data bank. Because we aren't using any personal structure, the software can handle everything by himself. If you are curious, you can check the processed structure, they should all be corrected from any seleno-methionines and have only one chain, the chain A or the one we specified.

It's time to upload our zip file to the mTM-align server! The zip file will be in the "Zip-File" folder inside your project folder.

On mTM-align, we can go to the multiple protein alignment section, and upload our zip file, then submit it.

Once the work is completed, we can paste the URL to the command fetch, in order to directly download results. Sometimes, the download can take several minutes, and is probably related to the server itself. If you don't want to wait, you can download them directly and place them in the "MSTA_results" folder

Make sure to put your project name as well, so it can place the results in the correct project.

```
> fetch example_MPSA https://yanglab.nankai.edu.cn/mTM-align/output/mTM013906/
```

The download is complete! We can start analysis now !

```
> analysis example_MPSA
```

And voila! The session is open and you can take a look at all similarities or differences within your structural alignment. It should look like the Figure 1.

Enjoy!