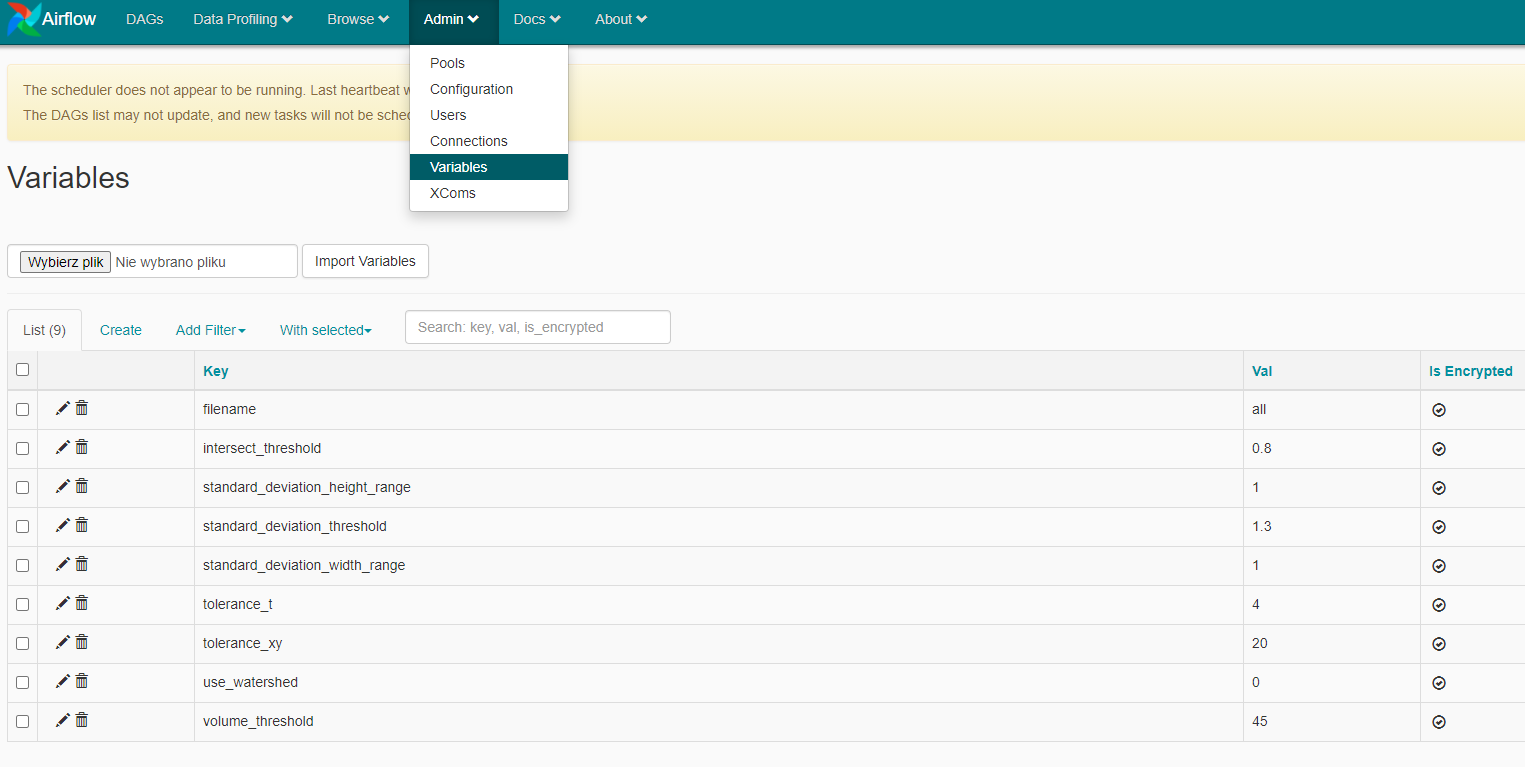
Astral – Manual

Astral is a tool for analyzing and interpreting calcium signalling from a microscopic timelapse imaging. The goal of Astral is to create an easily extensible system for such data.

There are couple of parameters available in Astral. You can see a full list of the parameters with their description below:

|  |  |  |
| --- | --- | --- |
| Filename | A filename of a timelapse sequence to be processed from the data directory. Leave ‘all’ if all the files should be processed | All |
| Intersect\_threshold | How much should the calcium waves overlap in their z-projection to be treated as repeats. | 0.8 |
| Standard\_deviation\_height\_range | How many adjacent pixels in height dimension to take into account during std calculation | 1 |
| Standard\_deviation\_threshold | Standard deviation threshold for detecting calcium events | 1.3 |
| Standard\_deviation\_width\_range | How many adjacent pixels in width dimension to take into account during std calculation | 1 |
| Tolerance\_t | A maximum distance between subsequent calcium waves to treat them as neighbours | 4 |
| Tolerance\_xy | A maximum distance in xy plane to treat two different calcium waves as neighbours | 20 |
| Use\_watershed | Whether during mask generation, a Watershed algorithm for splitting adjacent calcium waves should be used | 1 |
| Volume\_threshold | Volume threshold in total number of pixels to treat a shape as a calcium wave. Calcium waves below volume\_threshold will be discarded from the analysis. | 45 |
|  |  |  |
|  |  |  |

Those parameters can be set in the Variables section in the Admin tab:



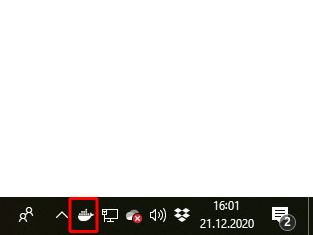
To change the parameter value, you may use a Pen icon next to the parameter name.

To change the default values of the parameters, you may modify the variables.json file. It is located in the root directory of the Astral application.

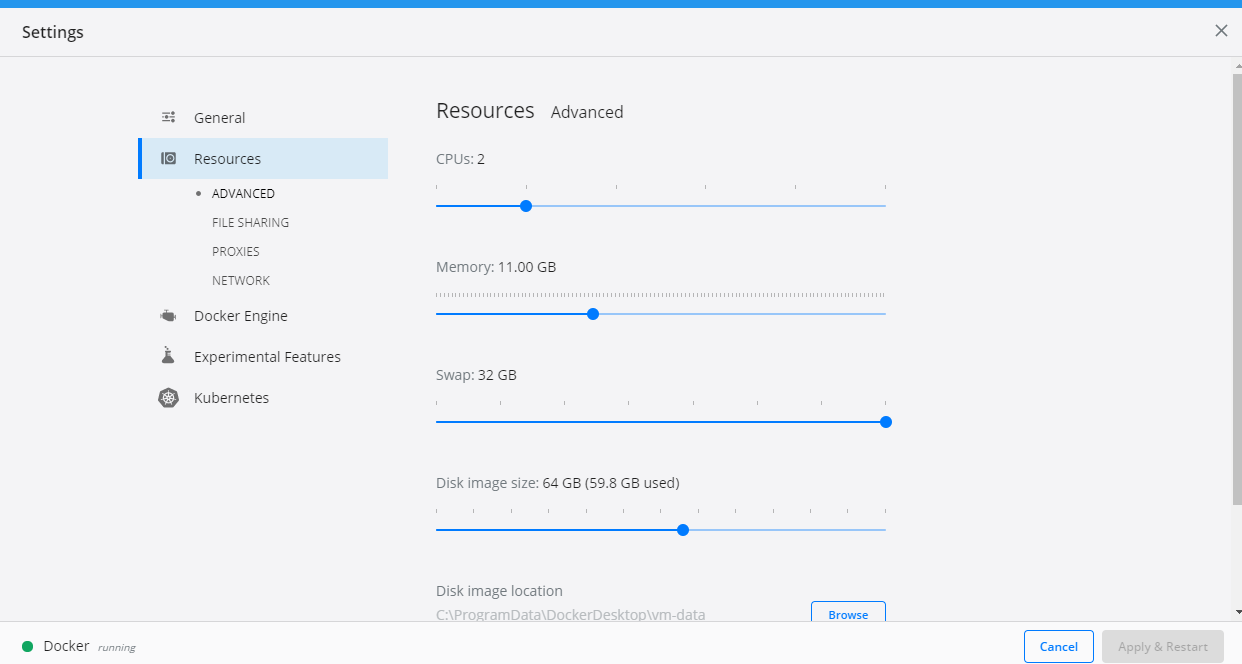
Installation process:

To run Astral, a Docker Desktop application is required. To download Docker Desktop, visit: <https://www.docker.com/products/docker-desktop>

After the installation process, you can launch Docker Desktop, and in the task bar, you should see Docker Desktop’s icon.



Right-click on it and navigate to Settings.



In the Resources tab, there are specifications for the computational power, that will be dedicated for Docker applications. You may manually customize, how much CPUs, Memory, Swap and Disk size you will give to Docker during runtime.

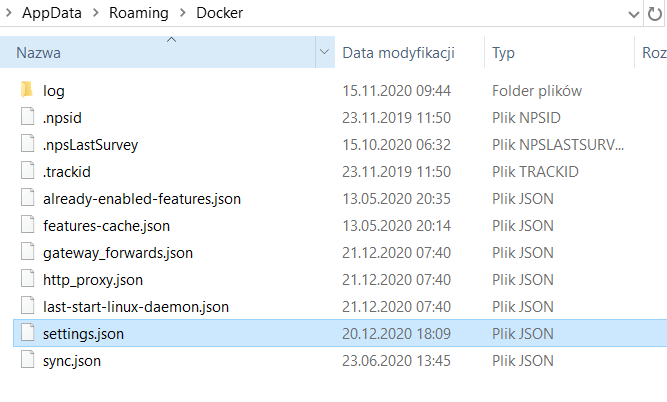
For Astral to run properly, a significant amount of Virtual Memory is necessary. If possible, set Memory to at least 10GB. For Swap, set it to 32 GB. Swap is a memory which is used when the application runs out of RAM. Because the timelapse data can be resource-consuming, it is advised to set it as high as possible.

Hint: If you are unable to set Swap to higher values, open up Windows Run window and type:

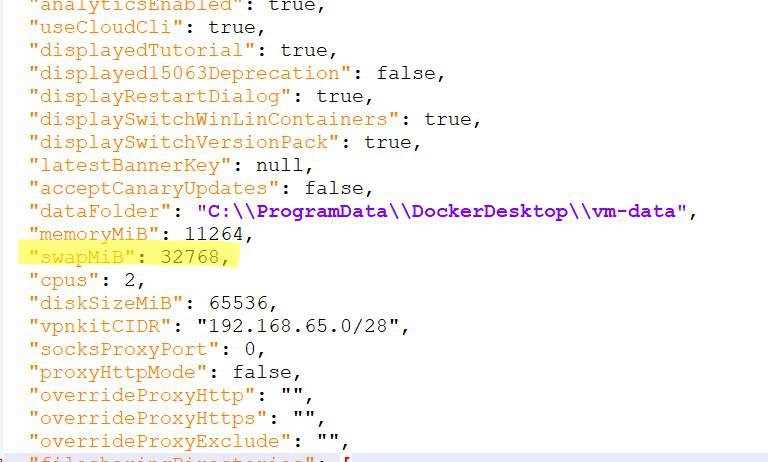
%appdata%

An explorer will show with an AppData folder. Then, navigate to the following file in the explorer.

AppData\Roaming\Docker\setting.json

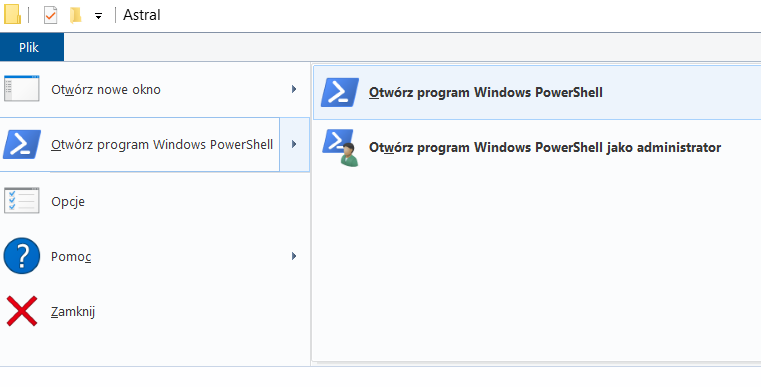


And open it with your favorite editor. Then, to set the swap memory manually, find the swapMiB property:



And set it to some preferred value.

To run the Astral application, navigate to the Astral application directory and from the explorer launch Powershell application:



First, you will need to build the application – for that to happen, type:

Docker compose build



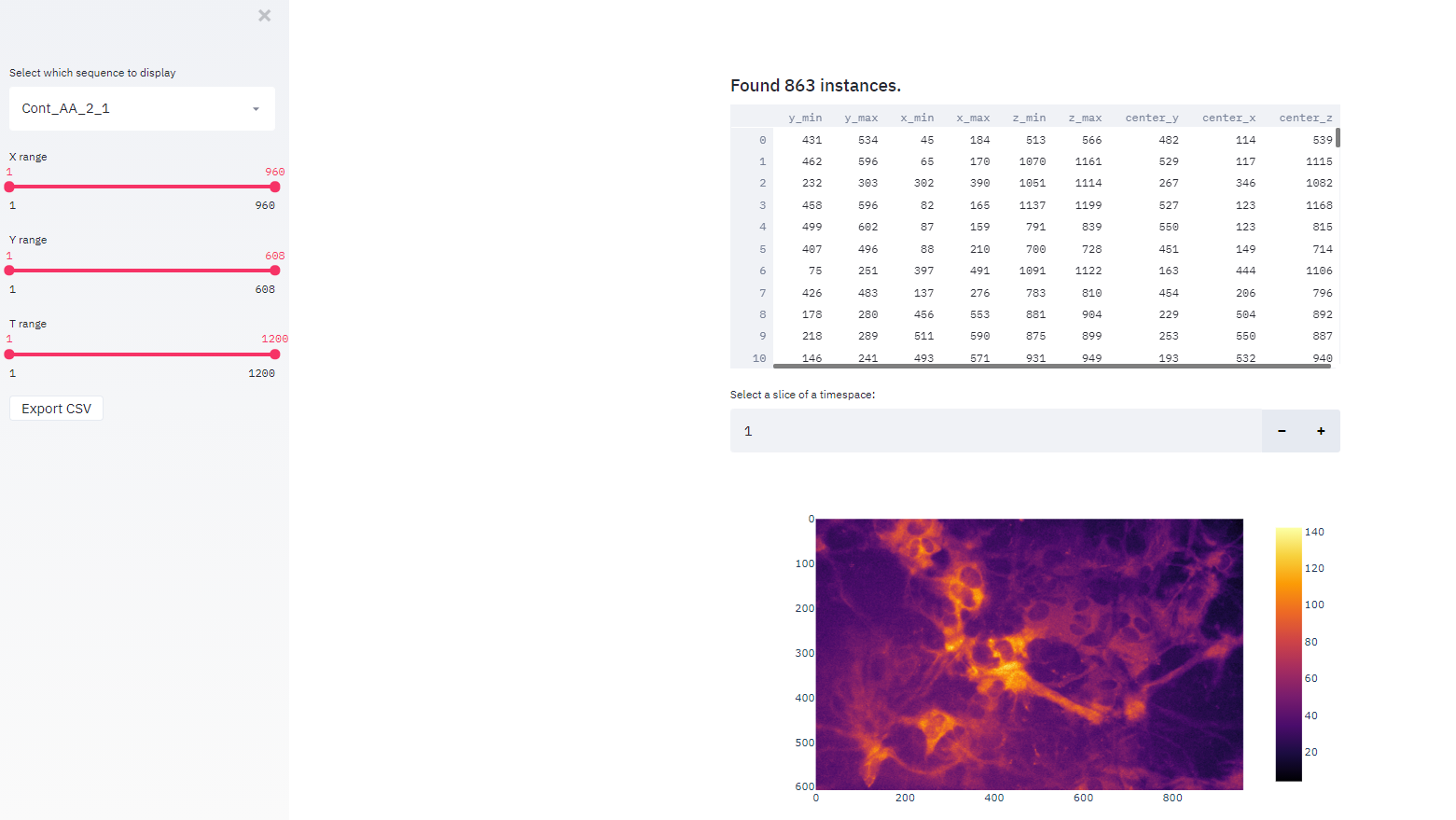
The application building process will start. The process can take up to several minutes, until the command prompt will be available for typing again. To start an application, type:

Docker compose up

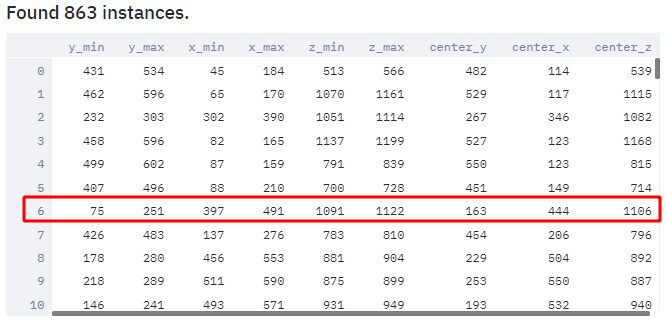
Then, in the Powershell window, the application log files will start to show. The running application will have the similar output at the window:

You can access the application on the URL address: http://localhost:8080/admin/ for Airflow application.

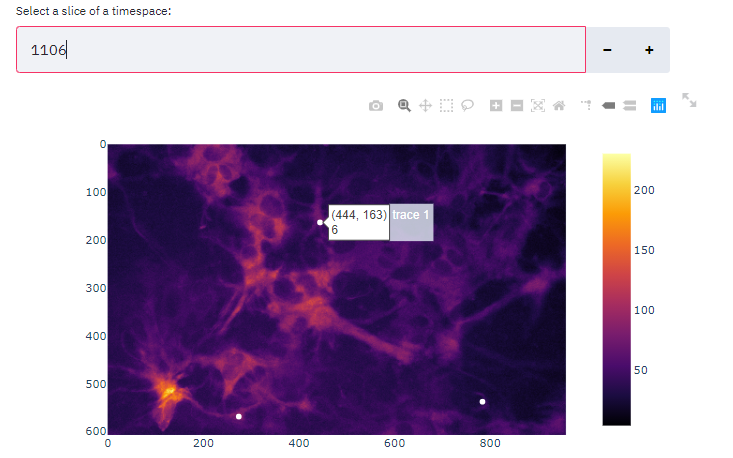
And on the URL address: <http://localhost:8501/> for the Streamlit application.



Streamlit application enables the user to inspect the timelapse in terms of detected calcium waves. First, a list of all the calcium waves with their ids and coordinates is shown. To check the location of a given calcium wave in the timelapse, it is possible to navigate through the timelapse. For example:

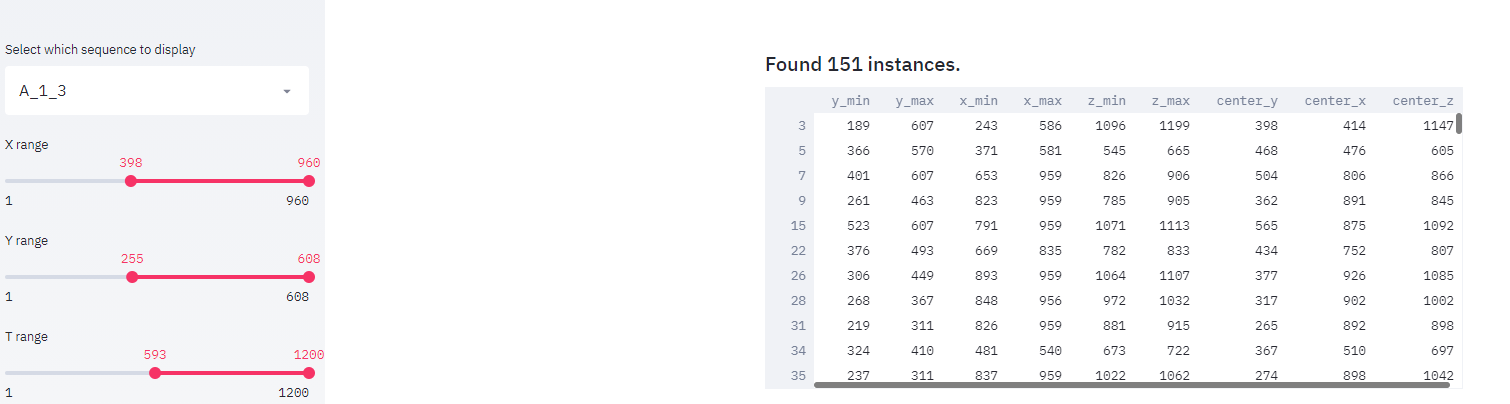


We would like to find the location in the series of the calcium wave with the id of 6. To do that:



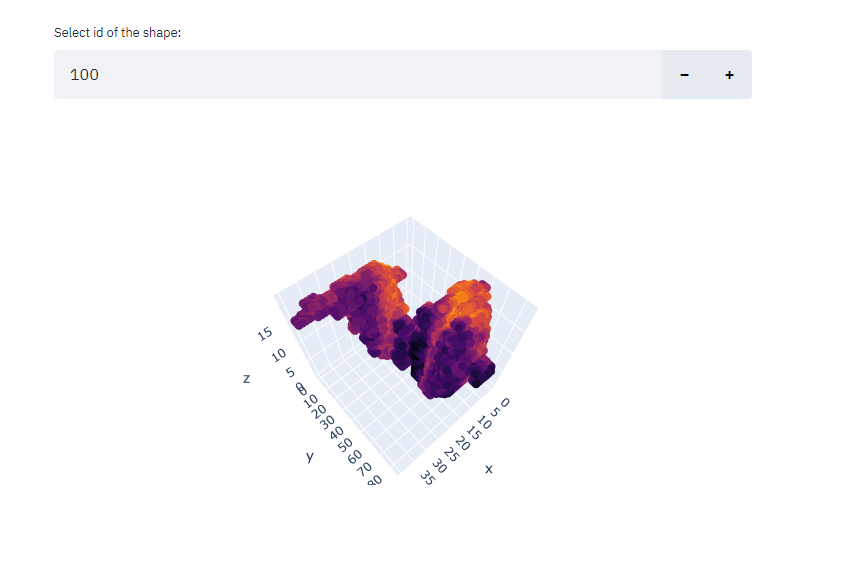
It is necessary to select a slice of a timespace of the center t coordinate of that calcium wave. Then, the calcium wave is marked with a white dot. In this case, two more calcium waves are also present on the slice.

Three sliders on the left can be used to filter the calcium wave list.



In this case, ranges in all the dimensions have been reduced. Only those calcium waves that are contained in this constrained subspace, are listed.

It is possible to view any calcium wave in 3D by selecting in by its id:



In this case, a calcium wave of id 100 is shown.