Documentation   
of the Pipeline for measuring Vessel Elasticity in a Sequence of retinal IR-Images  
By Viktor Dinkel, 13/09/2017

1. Software Requirements

* ImageJ with specific Plug-Ins
* Working Python Environment with all the different packages (see Source code for imported packages)

Optional:

* Apache XAMPP (or something similar like WAMP) for a Web- and MySQL-Server. This is optional but necessary if using the Database features of the Pipeline as well as the web-based results browser
* MySQL Workbench connects to the database and allows comfortable data inspection/manipulation

2. SetUp of the Pipeline

**ImageJ for image stabilization**

1. Get ImageJ from the Website: <https://imagej.nih.gov/ij/download.html>

This Plug-In was developed and tested on this downloaded version: bundled with 64-bit Java 1.8.0\_112 (70MB)

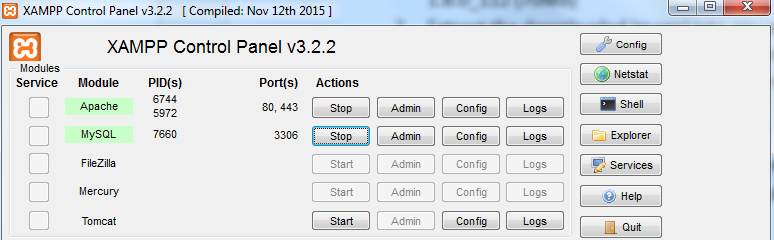
1. Extract the downloaded ImageJ into any working directory
2. Copy the contents of this folder directly into (your-ImageJ-directory)/plugins/
3. Run/Restart ImageJ, under the *Plugins* section should be the entry *disc stabilization*

**Working Python Environment**

This requires a 2.7 Python environment. There is no list of required packages, so the setup should be done with Anaconda or something similar with a package manager. The pulseWavePropagation code should simply be imported in any Environment.

**Apache XAMPP**

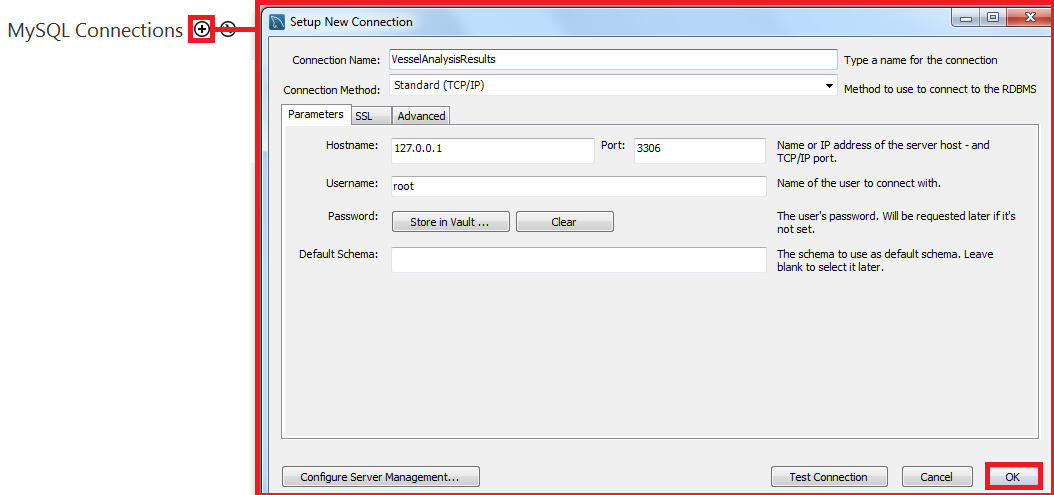
Download at <https://www.apachefriends.org/download.html> with the PHP version 5.6 for the corresponding platform. The installation routine is automated and no specific settings need to be done, just leave everything as default. After the installation is done, run the XAMPP Control Panel and start the Apache as well as the MySQL modules as shown here:



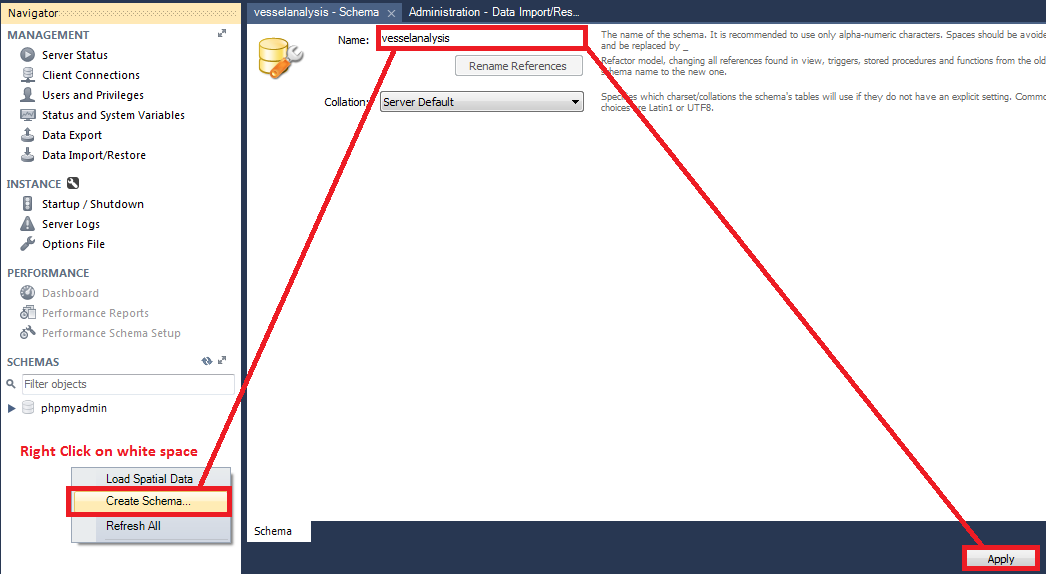
The launch of the XAMPP Control Panel and start of the modules has to be done after every restart of the machine (unless auto start was configured, see Config).

**MySQL Database**

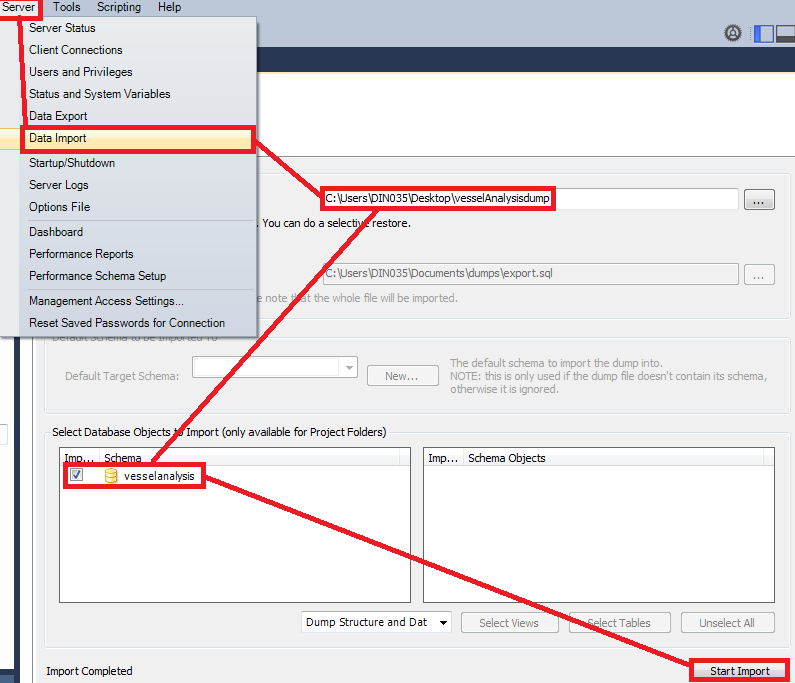
Can be downloaded here <https://dev.mysql.com/downloads/workbench/> and installed with all the default options. After installing and starting it, a new MySQL connection has to be created:



Click on the newly created connection and it should successfully connect to the database. Now the Schema needs to be created in order to import the tables into it:



Click *apply* and *finish* in the prompts, where the SQL-statement of creating the Schema is displayed again. In the final step, we want to import the tables with all the results which are contained in the vesselAnalysisDump.



In order to delete all containing results, sequences and individuals, these statements can be used:

delete from vesselanalysis.individuals WHERE 1;  
delete from vesselanalysis.results WHERE 1;  
delete from vesselanalysis.sequence WHERE 1;

Now the database with the necessary Schema is up and running and the Python-Pipeline can connect to it and insert/retrieve the vessel analysis results.

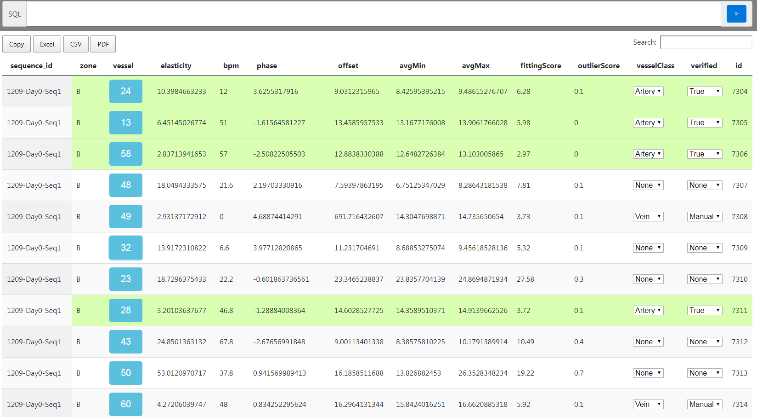
**Web based results browser**

Assuming XAMPP was installed in the default dir C:\xampp:

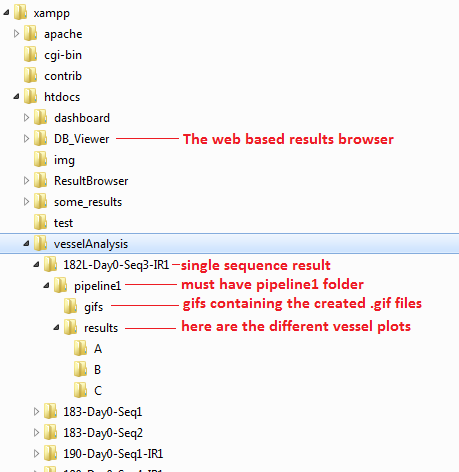
1. Copy the folder *DB*\_Viewer into the directory *C:\xampp\htdocs\*
2. Create a new folder *C:\xampp\htdocs\vesselAnalysis\*
3. Open your browser and enter this into the URL (requires running Apache and MySQL modules):

<http://localhost/DB_Viewer/results.html>

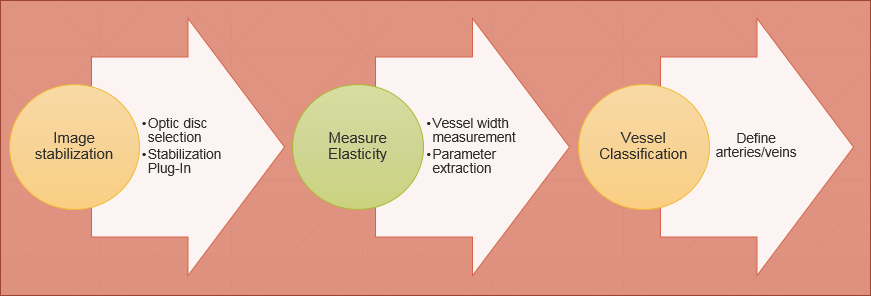
This should display the results browser and all the results from the results table. If the database is empty, then an error will occur. A reload of the page (and a database with results in it) will fix the displayed error.



In order to click on the blue vessel numbers and display the plots as well as the width measurement gifs, the results of the vessel analysis pipeline have to be copied manually into the created directory *C:\xampp\htdocs\vesselAnalysis\.* This is the structure which is necessary for that:



3. How to use the pipeline



Create a folder (*analysis folder*) where you copy all the separate sequence folders, which you want to analyse.

**Image stabilization**

1. Open ImageJ
2. Click *Plugins -> disc stabilization*
3. Select the first image of the sequence, which has to be stabilized
4. A frame opens up, click and drag a rectangle around the optic disc and click *ok*
5. A new folder pipeline1 is created in the same directory, where the image was selected. This folder contains the stabilized images as well as the disc parameters, which are used in the further processing during the pipeline
6. Repeat steps 2-4 for all sequences inside the *analysis folder*

**Measure Elasticity**

1. Set the *pathWithSequences* value as your path for the *analysis folder* like this:

*pathWithSequences = "C:\\Users\\DIN035\\Documents\\Project\\ ViktorDinkel\\SequencesToAnalyse\\"*

1. Set the value of the variable *pipeline\_step* in the file *IR\_VesselSegmentation.py* to *2* like this:  
   *pipeline\_step = 2*
2. Run the IR\_VesselSegmentation.py once. This will analyse all the folders inside your *analysis folder* and export the results to the database. If you don’t want to use the database, then change the value in the config as follows:

*Config.py -> GetConfigForMeasurement -> AnalysisSteps -> ‘exportToDB’: False*

All the results are saved in the corresponding \pipeline1\results\ folder of a sequence. The measurements are in the file classifiedVesselData.npy and measurements.npy (in the corresponding zones B and C) and can be loaded in the python code with the commands

*np.load(<pathToTheFile>+’measurements.npy’)  
np.load(<pathToTheFile>+’measurements.npy’).item()*

The vessels are not classified or verified yet.

**Vessel Classification**

1. Set the value of the variable *pipeline\_step* in the file *IR\_VesselSegmentation.py* to *3* like this:  
   *pipeline\_step = 3*
2. Run the pipeline once for each sequence (automatically iterates over the sequences for each new start of the pipeline)
   1. Select with the left mouse click a vessel spline to be artery
   2. Right mouse click selects the vessel spline as vein
   3. Middle mouse click resets the classification of selected spline
3. After the selection is done, close the opened figure window with the red cross in the top right corner, in order to continue with the pipeline and export of the vessel classification to the database

For this step, the database is required as the results are exported to it (updates the vesselClass in the database). In case the database is not used, then the function *DBActions.py -> updateVesselData* can be modified in order to process the selected arteries/veins in a custom way.

Furthermore, finished vessel classifications are stored in the *finished.npy* file and are skipped for further vessel classifications. If a reset is required (for re-classification), then the easiest way is to delete the file *finished.npy* although this also means, that all sequences within the folder, which have been classified, will have to be classified again if the code is run. A custom workaround can be programmed by manipulating the loaded *finished.npy* in the vessel classification step (adding/removing sequences).

This vessel classification step automatically sets the verified status of vessels, which have an outlier score <= 0.1 and were classified as arteries, to *True*.