**Remote Computing:**

**Great Lakes**

UM Slurm high performance cluster (HPC)

https://arc.umich.edu/greatlakes/

Host Name: greatlakes.arc-ts.umich.edu

Password: Level 1

**Galban Tier2**

Host Name: galban-ap-ps1a

Password: Level 2

A screenshot of a computer

Description automatically generatedAccess either through PuTTY or Remote Desktop. We’ve had some issues with RDP stability, so it is recommended to use PuTTY. For enabling of graphical interfaces, enable X11 forwarding (Connection -> SSH -> X11 -> Enable X11 forwarding)

**MySQL**

https://documentation.its.umich.edu/?q=node/380

Host Name: login.itd.umich.edu

Password: Level 1

***Database setup for Matlab:***

* Driver Location: This is where you download the Connector/J jar file from the MySQL website
* Input the rest of the values as shown in the screenshot to the right 🡪
* Save Source Configuration

**Third-Party Software:**

**YACTA**

**PuTTY**

SSH client for accessing Great Lakes, the tier2, and MySQL

Download: https://www.putty.org/

**Xming**

X11 forwarding program for use with PuTTY in displaying windows through SSH

Download: https://sourceforge.net/projects/xming/

**Elastix**

Software for image coregistration

Download: https://elastix.lumc.nl/

**TotalSegmentator**

GitHub: <https://github.com/wasserth/TotalSegmentator>

Install Python >= 3.7

Install Pytorch >= 1.12.1

Install TotalSegmentator:

pip install TotalSegmentator

Usage:

python TotalSegmentator -i ct.nii.gz -o segmentations

Options:

--ml : Saves segmentations in single file instead of individual binary files

From matlab:

system(‘cd /D <Save Path> & python <TotalSegmentator> -i <Input Nifti> -o <Save Path>’)

* <Save Path> = full path to where you want the results saved
* <TotalSegmentator> = full path to the python installation of TotalSegmentator
* <Input Nifti> = full path to the Nifti image to be processed

***For GPU enabling:***

Install CUDA v11.7 (see below)

Install Pytorch:

pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu117

Import Pytorch to Python environment:

>> import torch

Check if CUDA is available to Pytorch. In Python:

>> print(torch.cuda.is\_available)

Print Pytorch version in use:

>> print(torch.\_\_version\_\_)

**Pytorch**

pip3 install torch torchvision torchaudio

Documentation: <https://pytorch.org/docs/stable/index.html>

**CUDA**

Software for GPU control

https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html

* Requires an installation of MS Visual Studio
  + <https://visualstudio.microsoft.com/free-developer-offers/>
  + Visual Studio Community

**GitHub**

For updating the lab’s general Matlab software.

Download desktop app: <https://desktop.github.com/>

OR simply download code from: <https://github.com/hoffba/cmi_R2015a>

**Screen (for use on the Tier2)**

Allows running processes that are not killed when you close your SSH connection.

Start a screen: >> screen -S screen\_name

Detach from a screen: Ctrl+A, Ctrl+D

Reattach to a screen: >> screen -r screen\_name

List existing screens: >> screen -list

Kill a screen: >> screen -X -S screen\_PID quit

**Not Working**

**Windows Subsystem for Linux (WSL)**

This is a virtual Linux workspace.

In cmd.exe as administrator: wsl –install

**Docker Desktop**

Provides a GUI for managing containers, using WSL to isolate environment versions for certain applications.

**NVIDIA PyTorch Release Containers**

<https://docs.nvidia.com/deeplearning/frameworks/pytorch-release-notes/rel-23-03.html#rel-23-03>