Setting up and installing the GEO244 conda environment in WSL2

Gareth Funning (gareth.funning@gmail.com)

In order to make use of the various teaching materials produced for the InSAR training course, you will need to install certain software packages.

This assumes you have already set up a functional WSL2 instance on your computer (e.g. via the instructions linked here: http://www.geniusinaction.com/2023/01/wsl2-in-minutes.html).

1. Install the miniforge package installer

Miniforge (https://github.com/conda-forge/miniforge) is an open source package installer for Python. It is similar to the commonly-used Miniconda package installer, except it is faster and community-maintained, rather than maintained by a commercial company (the Anaconda company maintains Miniconda).

For WSL2 users, the 'x86_64' (amd64) version for Linux is the correct one to install. You can use this download link:

https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh

To actually install it, you will need to run this script in WSL2. To do that, it is probably best to build a symbolic link to your Windows 'Downloads' directory in your WSL2 home directory. You will need to know what your Windows username is to do this. Mine is 'Gareth' and so I would use this command:

```
ln -s /mnt/c/Users/Gareth/Downloads/ .
```

(You should substitute your own username here instead of 'Gareth'.)

Hopefully now when you do a directory listing, you should see a 'Downloads' directory listed there…

```
ls
```

```
Downloads
```

Now you can change directory to your Downloads directory and run the Miniforge install script:

```
cd Downloads

sh Miniforge3-Linux-x86_64.sh
```

and you will have to accept the terms and conditions and approve a few things (e.g. the install location, whether to add some terms to your .bashrc script) – the defaults are all fine.

Once this is done, close your WLS2 terminal and reopen it.

In the new terminal, if you did everything right the user prompt should look something like this:

```
(base) gareth@crouch:~$
```

The '(base)' in the prompt here refers to the conda environment you are currently working in – in this case, the base environment. We will want to set up a new environment with specific packages in it for the training course.

2. Download and use the InSAR.yml file

I have posted a YAML file, 'InSAR.yml' to my GitHub site for this training (https://github.com/geniusinaction/InSAR_training). It contains a set of software packages that we will use in this course. If you managed to successfully install Miniforge, the next step is to use Miniforge to install a conda environment using that file.

Using your web browser, download the file located at this link:
https://github.com/geniusinaction/InSAR_training/blob/main/InSAR.yml

Assuming that you downloaded this to your Downloads directory again, navigate there.

```
cd Downloads
```

Now we can use the 'mamba' command to download and install all of the packages we need to run InSAR things:

```
mamba env create -f InSAR.yml
```

And now we can activate the environment we just set up using the 'conda' command:

```
conda activate InSAR
```

3. Set up a .netrc file with your login details

In some of the data processing workflows we may use, it is useful to have your Earthdata and Dataspace login details in a file ('.netrc') where the software can find it. We can use the nano text editor in the terminal to make and edit that file:

```
nano .netrc
```

And in the text editor that opens, type the following (**substitute in your own login details and passwords**, obviously):

```
machine urs.earthdata.nasa.gov
login my_earthdata_username
password my_earthdata_password
machine dataspace.copernicus.eu
login my_email_address
password my_dataspace_password
```

You can save and exit nano by hitting Control-X and then y

One more thing: you need to set permissions so that only you can read this file:

```
chmod 0600 .netrc
```

And then you should be good to go!

4. (For ISCE) Make a script with path details

For the major workflow scripts of ISCE (topsApp.py, stripmapApp.py, etc), you do need to set up the paths to those things properly.

I usually make a file called ISCE-config.sh and put some configuration details in it. First open the file:

```
nano ~/ISCE-config.sh
```

And in the text editor that opens, type in the following:

```
export PATH=$ISCE_HOME/applications:$ISCE_HOME/bin:$PATH
```

You can save and exit nano by hitting Control-X and then y

One more thing: you need to execute this script to get it to work:

```
source ~/ISCE-config.sh
```