# 1 Required and suggested software for the course RSM02

## 1.1 Objectives

This document briefly describes what Software to setup for performing analysis steps and working with point cloud data and for lidar classification. *Please install this software on your laptop - it will allow you to use it locally and if required, you can access the PC Pool (logins will be provided). All software is open source.*

You can use Windows and most of the software will also be easily installed on a Mac. *We suggest to use Ubuntu or some other Linux-based distribution as these are the most flexible systems, but Windows OS will work as well.*

# 2 Software

## 2.1 Command Line Tools and GUI

Packages only used for pointcloud data:

- CloudCompare
  - Point Cloud analysis and visualization. Includes many useful point-cloud analysis tools, but is slower for the visualization of large pointclouds. *Runs on every OS.*
- displaz
  - Very fast and versatile viewer. Can be run from python, but we mostly use the command line.
  - *Windows Users:* Download the binary and run.
  - *Ubuntu Users:* This likely will need to be compiled on your machine - follow the instructions on the github page
  - *Mac Users:* Due to the recent updates for the X11 Server, this doesn't properly compile. You may end up using CloudCompare instead (which is slower for visualization, but still allows you to do all steps).
- LAStools/LASlib
  - Open Source Software for working with LAS files.
- LAStools

  - Commercial Software. Very powerful and fast. Licenses are available on the PC Pools (logins are provided). You can install an unlicensed version on your computer and run some of the commands locally.

We expect you to install the packages CloudCompare, displaz, and LAStools on your own. Installing instructions are provided on the download webpages. LAStools/LASlib will be installed via anaconda/miniconda (see next).

## 2.2 Python Packages

We will rely on the following *python* packages and environments as well as several tools for pointcloud analysis, some python programming, and general data analysis:

- Python 3.x
- PDAL
- LAStools/LASlib
- GDAL
- scipy
- numpy
- pandas
- pylidar
- laspy
- and several other tools

### 2.2.1 *Windows Users*: Install command line tools and Python packages

One option is to install this via Anaconda and select the packages *gdal, pdal, Pylidar, pdal, lastools, numpy, pandas and matplotlib*.

You can also install the required packages via the `anaconda shell`. Depending on your installation, you may need to add the channel *conda-forge* to the search environment:

```
conda config --prepend channels conda-forge
```

I suggest to create a separate conda environment dedicated to the analysis of pointcloud data (e.g. `Py3_PC`):

```
conda config --prepend channels conda-forge
conda create -y -n Py3_PC python=3.* pip scipy pandas ^
  numpy matplotlib scikit-image gdal ipython spyder ^
  statsmodels jupyter pyproj lastools pdal ^
```

```
  pykdtree h5py
conda activate Py3_PC
pip install laspy
```

**Alternative option Windows Users** Install Linux Subsystem on Windows and use miniconda (see next section). Installing the Linux subsystem (use Ubuntu 18.04) is generally a useful thing to do for Windows users (if your hardware space allows it), but is not required for this course.

#### 2.2.2 *Ubuntu and Mac Users*: Install command line tools and Python packages

You can install Anaconda for Mac, but you may prefer the command line approach described below. Install miniconda3 and the packages via `conda install`. Download and install the required software via the command line/shell:

```
cd ~
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
sh ./Miniconda3-latest-Linux-x86_64.sh
```

You may have to include additional channels for installation:

```
conda config --prepend channels conda-forge
```

Install the conda packages (will take some time):

```
conda config --prepend channels conda-forge
conda create -y -n Py3_PC python=3.* pip scipy pandas numpy matplotlib scikit-image  \
    gdal ipython spyder statsmodels jupyter pyproj lastools pdal pykdtree h5py
conda activate Py3_PC
pip install laspy
```

## 3 Additional considerations

### 3.1 Editor

We will be doing some coding and it may be useful to use an editor to take notes as well. Install your favorite editor - for example Atom or Notepad++ on Windows or Spyder. Spyder is included in the Windows Anaconda distribution and is installed via the command line above.

# 4 Remote login to the PC Pools in Building 27 in Golm

The remote login to the PC Pool computers will work through a ssh connection. This command-line based approach allows you to run the commercial software package LAStools and will allow you to run calculations for a longer duration of time. You copy data from your local computer to the remote computer via sftp or rsync on Mac/Ubuntu, robocopy on Windows-PowerShell, both are for the command line, or FileZilla (use the Filezilla Client) (GUI). If you are not familiar with the command line, I suggest to use FileZilla, which supports the sftp protocol and drag'n'drop procedures.

*Keep in mind: Likely, your download speed will be very fast (i.e., it will be fast to pull files from the PC Pool), but your upload speed will be slow (i.e., you have to be patient if you put data on the PC Pool)*

**You will be sent a separate Email with login information to the PC Pool computers. There will be some sharing necessary because there are more participants than computers, but this will barely effect computing efficiency.**

## 4.1 Login via ssh

SSH stands for secure shell and is a safe and encrypted way to access computers remotely. X-Windows (graphical interface) commands can be also forwarded, but this is very slow and not suggested. *We will work with the SSH command line environment*. SSH is implemented on all OS.

### 4.1.1 Windows Users

You need to start the PowerShell (From the Start Menu Click Start, type PowerShell (search for it), and then click Windows PowerShell). You can simply login in to the PC pool computer with (I am using the IP of `pcpool1`):

```
ssh student@141.89.113.160
```

Replace YOURUSERNAME with your login name sent to you (`student` or `student1`) via email and use the IP address 141.89.XXX.YYY given in the email as well. Your login will only work on that system assigned to you.

### 4.1.2 Ubuntu/Linux and Mac Users

Open the terminal and use ssh. You can simply login in to the PC pool computer with (I am using the IP of pcpool1):

```
ssh student@141.89.113.160
```

Replace YOURUSERNAME with your login name sent to you (student or student1) via email and use the IP address 141.89.XXX.YYY given in the email as well. Your login will only work on that system assigned to you.

### 4.2 SSH Note

If you are interested in a graphical login (i.e., you can open windows from the command line), use for example:

```
ssh -X student@141.89.113.160
```

But be prepared - this is very slow and not advised. It may take several minutes until a graphical window opens.

*You can use the ssh login and work on the remote computer - similar to your terminal / shell on your local computer.*

## 5 Data storage on PC Pool computers

Do not store your data in your home directory (`/home/student`). Instead use `/DATA` for fast, local storage and `/NAS` for network storage. I suggest that you create your own directory. For the user bodo, I would write (you should use your own name ;):

```
mkdir /DATA/bodo
```

*We will periodically wipe/erase the home directory and you would loose all your data.*

## 6 File transfer between local and remote computers

### 6.1 FileZilla (Graphical User Interface) - *Your most likely choice*

If you don't want to use the command line, you can install Filezilla Client and create a directory entry for your remote PC pool. This is a straight forward way to copy files back and forth (you see progress and expected up/download times). This will work well for a few files. If you plan to move back-and-forth files continuosly, you may want to look into rsync or a similar

### 6.2 Command Line option 1: rsync (only for Ubuntu/Mac Users)

If you like to work on the command line (because it is more efficient), rsync is the best option.

### 6.3 Command Line option 2: scp (only for Ubuntu/Mac Users)

With scp you can copy files from the command line. The syntax is simple: `scp <localfilename> \ <remotefilename>`. For example, if I (user: `student` on computer `141.89.113.160`) want to copy the file `my_python_script.py` from my local computer to the pc pool, use:

```
scp my_python_script.py student@141.89.113.160:/DATA/bodo
```

### 6.4 Command Line option 3: sftp

The local computer is your own laptop/desktop computer, the remote machine is the computer in the PC Pool. With sftp you can copy a file on the command line (*Windows Users:* Use the powershell, *Ubuntu and Mac users:* use the terminal/command line). The syntax is simple: `sftp \ <remotecomputername>`. For example, if I (user: `student` on computer `141.89.113.160`) want to copy the file `my_python_script.py` from my local computer to the pc pool, use:

```
sftp student@141.89.113.160
cd /DATA/bodo
put my_python_script.py
```