# Accessing the workstation

You need:

A program where you can access the terminal via a command line. There are many different programs for this, but I use Visual Studio Code. Here you can also copy files from the workstation to your own computer and read and write files and scripts.

A computer screen capture

Description automatically generated with medium confidence

Terminal, this is where you write commands and start programs

Here you write scripts, and check readable results files

More easily readable folder list

1. Download VScode

<https://code.visualstudio.com/>

1. Install git bash (you need the program to be able to read shell bash files, which is the programming language for linux

<https://git-scm.com/downloads>

* 1. Select bash as the default language for the terminal

<https://www.shanebart.com/set-default-vscode-terminal/>

1. Install the extension ssh FS (to connect to the workstation via a ssh connection)

<https://marketplace.visualstudio.com/items?itemName=Kelvin.vscode-sshfs>

1. Configure connection to workstation
   1. A screenshot of a computer

      Description automatically generated with medium confidence
   2. Fill in these fields:
      1. Name: what you want the connection to be called. Can be anything “eg. Workstation”
      2. “Host”: type in the IP address of the workstation: **10.11.200.20**
      3. “root” this is which directory you want to connect to. I’d choose /data01/$USER
      4. “username”
      5. “password” – choose prompt. You will have to type it in every time you want to log in. It is not recommended to put the password in the configuration file.
   3. If you are not working connected to Scientific Network SouthTyrol, you need to access it first through the University VPN
2. Log in
   1. Directly via terminal:
      1. $ ssh -X [$USER@10.11.200.20](mailto:$USER@10.11.200.20)
   2. Via ssh fs
      1. Add as workspace folder
      2. A screenshot of a computer

         Description automatically generated with medium confidence
      3. At prompt, put in password
      4. Right click on the Workspace folder and click on “open remote SSH terminal”. Will not require a password prompt, since you already typed it in when you logged in to workspace folder
      5. A screenshot of a computer

         Description automatically generated with medium confidence
3. First time you log in – change the password!
   1. $ passwd
   2. Graphical user interface, text

      Description automatically generated

User storage

* Every new user has a directory (folder) in /home/$USER
  + this has limited storage, it is only used for configuration files.
  + DON’T DO ANY ANALYSES IN THE /HOME FOLDER
* every user also has a directory in /data01/$USER
  + this is where you store your data, scripts, log files, intermediate files and results
  + for now unlimited, but if we are many that uses the workstation there will be a limit of 3-4TB of data storage per person
* you are only supposed to be able to write files in these two directories, if you can do it somewhere else, let Systemadmin (Lena) know immediately!
* The files are backed up daily, so if you delete something irreplaceable (like raw data not stored somewhere else or files created in a stochastic way and can’t be reproduced) let sysadmin know. Generally things that will generate the same output and done quickly you don’t need to restore, just rerun.
* In order to save space, use symbolic (soft) links to eg. raw data.
* Always compress big files to save space!
  + Fastq files to fastq.gz (most software can work with this, otherwise use zcat to pipe the compressed file through a command without decompressing it first ($ zcat file.gz | software -options >out.example).
  + same with bam files, which are binary version of sam files (alignment files). Also, sorted bam files take up much less space while compressed, so use samtools sort to sort them

The Linux terminal

To learn how to navigate the linux terminal, please do a tutorial for the basic commands. For example how to move through the directories, copy, move and delete files, list directories and files and so on.

<https://ubuntu.com/tutorials/command-line-for-beginners#1-overview>

Running bioinformatic analyses and scripts with conda

* All software is installed by the System Administrator. If you want a new software downloaded, please send Lena an email.
* Software is run through the environment manager program called Anaconda3. you therefore have to “load” a software before you can use it:
  + $ conda activate bowtie2
  + $ bowtie2 sample.fastq reference.fa
  + $ conda deactivate bowtie2¨
* To list all environments/software installed on the workstation:
  + $ conda list

File transfer

You can transfer files to/from the workstation through VScode, but a quicker and more controlled way is through a file sharing program like filezilla or WinSCP. Access the workstation with the same configurations as for VScode.

Use of computing resources

* CPU/threads and RAM memory
* To speed up your jobs bioinformatic software usually have options that specify how much resources to use. There is usually a default value, but software can run many times faster with a higher CPU usage. Check in the options of the software, it is often specified with -t or --thread but can be something different

Tips and tricks

* Use the tab button to auto complete the name of a file or directory in the terminal
* Ctrl+arrow button will jump through separated fields in any text, including the terminal (also in word files)
* For loops are your friend. Instead of running the same analysis on different samples one after another, create a loop that will automatically start one job after the first one is finished
* For all that is holy, remember to make comments in your scripts. You will not remember what a command is for or how to run it after a few months.