Setting up your computer environment

General information

One of the main challenges in bioinformatics analyses - though frequently neglected - is the set up of the computer environment such that you can run your analyses smoothly. Rather than providing you with a nice and shiny pre-installed environment, we will guide you through the main steps of setting up your system on your own. Please note, installing software for comparative genomics works quite well for LINUX and MacOS, but if you run Microsoft_Windows you are likely to encounter considerable issues. This is, since many of the software for biological sequence analysis has not been implemented for this operating system. In this case, you may want to consider installing a LINUX distribution on a virtual machine.

The conda package manager

Whenever possible, we will use the conda package manager to install the software we need for the individual analysis steps. The main advantage of this package manager is that it not only installs the program you wish to use, but also all the dependencies you are often not aware of. The conda package manager, which can be easily installed on your system provides access to the Anaconda cloud. Using the conda package manager is considerably straightforward and we will guide you through the first steps. However, reading the documentation and using cheat sheets can certainly not harm. Sometimes, it may also help to take a look at the Anaconda glossary to get the meaning behind some of the terms used in the context of conda.

⚠ We will be using **Miniconda** in the course, this is sufficient for our purposes

Mamba

The standard installation of Anaconda becomes considerably slow after a couple of different environments have been set up. We therefore recommend using Mamba which is much faster than Anaconda but otherwise behaves just as Anaconda does. In fact, Mamba is build upon Anaconda so installing Mamba will also install a bare bones version of Anaconda.

After installing mamba, you can replace the `conda` command you use for creating environments and installing packages with `mamba` to drastically improve speed. Throughout this wiki, the `conda` and `mamba` commands will be used interchangeably. You are free to use either one, but `mamba` will be faster most of the time.

Setting up Mamba

- You can follow the installation guide on the official Mamba documentation page or refer to the steps below

1. Download the latest `x86_64 Linux` installer from the miniforge GitHub page

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

   1. Start the installation by typing

      ```
      bash Miniforge3-Linux-x86_64.sh
      ```

      1. The installation process will ask you to read and agree to the license agreement. Press ENTER to see the agreement, then scroll down. At the end you can agree to them by typing **yes**. When asked in which location Anaconda should be installed you can just press ENTER to accept the default location (e.g. /home/your_name/anaconda3).
      2. Towards the end of the installation, the installer will ask you:

         ```
         Do you wish the installer to initialize Anaconda3 by running
         conda init? [yes|no]
         ```

         Answer with:

         ```
         yes
         ```

         This directly activates conda's base environment whenever you open a terminal.

      3. If you wait for some time, Anaconda will automatically answer with “no”. In this case, please activate Anaconda manually with these two commands:

         ```
         source <path to conda>/bin/activate
         conda init
         ```

4. At the end the installation process will tell you

   ```
   For changes to take effect, close and re-open your current
   shell.
   ```

   So close your terminal and open a new one.

5. Check if the set up was successful:
   1. In the new terminal that you have just opened, it should say something like:

      ```
      (base) francisca@workstation1:~$
      ```

      This “(base)” indicates the base conda environment is activated. Now type:

      ```
      conda -h
      ```

      This should print some basic help to your terminal. If all is good, proceed to the next step.

6. **Creating a Conda environment.** We will now create a working environment for your analyses. It has the advantage of keeping software that you installed via the package manager together in one spot[1]. Moreover, it helps to have different - and even conflicting - set-ups at the same time on your computer. For example, you can have two environments using different versions of python, and you can easily switch between them.
7. To create a working environment named ***compgen*** with python 3.9[2], type

   ```
   mamba create --name compgen python=3.9
   ```

8. To activate the environment, type

   ```
   conda activate compgen
   ```

9. To leave the environment, type

   ```
   conda deactivate
   ```

   ⚠ If you want to switch between environments, please **de-activate** the current environment, and only then activate the new environment!

10. To remove an environment permanently, type

    ```
    conda env remove -n EnvNAME
    ```

    ❗ Please replace *EnvName* with the name of your environment

11. To print out some information about your conda installation to your screen type

```
conda info
```

into a shell. You will obtain something like the following:

```
active environment : compgen
active env location : /Users/ingo/anaconda/envs/compgen
shell level : 1
user config file : /Users/ingo/.condarc
populated config files : /Users/ingo/.condarc
conda version : 4.5.11
conda-build version : 2.0.2
python version : 2.7.12.final.0
base environment : /Users/ingo/anaconda  (writable)
channel URLs : https://conda.anaconda.org/conda-forge/osx-64
```

12. If the installation was successful, you can ask for further help with the individual commands,such as, *e.g.* ***install*** , ***update*** , ***remove*** by typing

    ```
    conda install --help
    ```

13. Do not install anything in your base environment. This might cause Mamba/Anaconda to malfunction. Always create a new environment and install packages in there
14. You are now ready to install any software that is distributed via the conda package manager using the following command

    ```
    mamba install CHANNELNAME::PACKAGENAME
    ```

15. The software packages are distributed via different channels. We will mainly use the channels bioconda and conda-forge.
16. To simplify your life, it might be a good idea to add frequently used channels to a list of channels where conda searches for your package by default. Use the following syntax to add the channels bioconda and conda-forge to your default channel list

    ```
    conda config --add channels bioconda
    conda config --add channels conda-forge
    ```

If you completed all of the steps above, then you should be ready to make full use of the conda package manager.

[1] There is a sweet spot about the package manager, which you should be aware of. Each time you install a program, conda will ask you if it is ok to install, update, or sometimes downgrade a number of packages. This can lead to almost unforeseeable incompatibilities between different programs installed in the same environment. To avoid this, some people go as far as creating for each program its own environment…

[2] In case you require both python 2.7 and 3.9, then simply generate an environment for each python version

From:
https://fsbioinf.biologie.uni-frankfurt.de/teaching/wiki/ - **Teaching**

Permanent link:
https://fsbioinf.biologie.uni-frankfurt.de/teaching/wiki/doku.php?id=general:computerenvironment:conda

Last update: **2024/04/15 12:45**