

Chapter 1

Installation

These instructions are needed for exercise 6 and 7 to work. Please run these instructions before going to the web-based environment of CACATOO for question 0-5 so we know the software in later weeks will work! These instructions were originally made to install all the dependencies for a cellular automaton simulation programme called Cash2.1 on Linux, or on Windows (using Windows subsystem for Linux and Ubuntu), or on MacOS. Some of the dependencies might not be needed for what we now run on Linux (chiefly an RNA-evolution programme), but please install them all for now just to be sure. The general installation process is the same for all systems:

- Installing dependencies (libraries)
- Installing software for display and graphing

1.1 Installing on Linux (own laptop / desktop)

Installing the required dependencies can be done manually (see blue box below, only for cool kids) or by using the script provided on our course website (preferred, easy option!):

- `wget https://tbb.bio.uu.nl/BINF/software/Cash2.1/cash_dependencies.sh` . **NOTE:** if you copy this into your terminal window the `_` is often not copied along. Be sure to include it manually, or just download the file `cash_dependencies.sh` from the linked folder in your browser!
- `sudo ./cash_dependencies.sh`
- (optional) If that doesn't work right away, the file might not be marked as an executable. To fix this, type `chmod a+x cash_dependencies.sh`. Then retry the step above and continue.
- (enter your password)

For geeks that want to do it themselves, here's what the script does:

- `apt-get update`
- `apt-get install unzip`
- `apt-get install build-essential`
- `apt-get install grace`
- `apt-get install x11-apps (just for testing)`
- `apt-get install libpng-dev`
- `apt-get install zlib1g-dev`
- `apt-get install libx11-dev`
- `echo "export DISPLAY=:0.0" >> ~/.bashrc`
- `source ~/.bashrc`

To edit programs written in a language like C, such as you'll be using in exercise 6, it is preferable to use a code editor. A good choice is VScode. Get it here: <https://code.visualstudio.com/Download>. Alternatives are Notepad++ click, or Atom click. For VScode, be sure to download the C/C++ extension. You can of course edit the code with whichever application you wish, but these programs help you by highlighting important things, suggesting autocompletions, etc. Be sure to search for handy keyboard shortcuts in VScode or another editor!

1.2 Installing on Windows

Since Windows 10, the Microsoft app store has 'Ubuntu for Windows' freely available, which is a small, yet fully functional, Linux Ubuntu distribution run on windows. By installing this, and installing a program to manage your display (displaying graphs etc.), you can use the software in this course without any limitations!

Step 1/5: Preparing Windows 10 developer mode

- Click on the Start menu → type 'developer settings' → Enable the 'Developer mode'
- Go to Control Panel → Program's and Features → Turn Windows features on or off (top left corner). Enable 'Windows Subsystem for Linux' (see Figure 1.1)
- (Restart Windows)

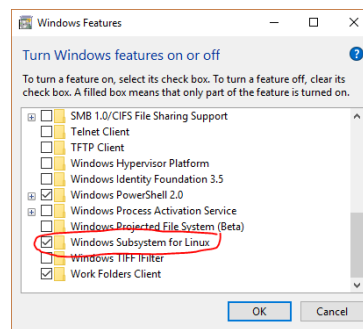


Figure 1.1: Enable Windows Subsystem for Linux

Step 2/5: Installing Ubuntu for Windows 10

- Install Ubuntu for Windows 10, available in the Microsoft app store.
- Open this file (<https://tbb.bio.uu.nl/BINF/software/WindowsUsersSetWSLToVersion1.pdf>) and follow the steps there to make sure you have WSL 1, rather than WSL 2 which is used by default. It's a tad slower but it's much easier to get Linux to output graphics!
- Start Ubuntu for Windows. Continue through the installation. (setting up a username and password, etc.).
- By default, your Windows documents can be accessed via `/mnt/c/Users/<username>`.
(note: it is advised to download and install software you need here, since you can then easily edit the code via your preferred Windows text editor, without getting permission problems)

- We recommend installing VScode, which is a powerful IDE (integrated development environment) that you can easily start from Ubuntu if you want. Get it here: <https://code.visualstudio.com/Download>. Once opened, it will detect that you have WSL and ask to install an extension. Do this. If you open a .c file, it will also prompt you to install an extension to work with C code. Allow this too. You can open VScode from the WSL by simply typing *code*. Try it!
- If you don't want to use Visual Code, a simple alternative is Notepad++ click, or Atom click. Whichever you choose, be sure to search for handy keyboard shortcuts online!
- To make your life easier, right click on an opened Ubuntu terminal at the top, go to properties, and allow copy-pasting with `ctrl+shift+v` and `ctrl+shift+c` and enable quick edit mode. In this way, you can click-and-drag to select lines of code, and use copying and pasting easily! You can also change the look of your terminal here, if you want.

Step 3/5: Installing an X server for Windows

In order to run graphical applications via your newly installed Ubuntu, you need an X server. There's a small one called VcXsrv available here sourceforge.net/projects/vcxsrv/

- Install this program
- Restart Windows
- Launch the program. If it asks you to select modes and options, simply click through, accepting all defaults. You may need to do this every time upon startup (but see NOTE! below). After you've clicked through you'll see nothing on the screen. That is correct! All this program does is show you graphical output from the Ubuntu terminal. Since your terminal is not currently making output, you can only see that it is running by an icon in your system tray (bottom right small icons on your screen).
- Open Ubuntu and copy-paste this command: `echo "export DISPLAY=localhost:0.0" > ~/.bashrc .` And then `source ~/.bashrc .` The result should be the line "export DISPLAY=localhost:0.0"(without the quotes) in the file `~/.bashrc`. You can test this by typing `nano ~/.bashrc` scrolling down and seeing what's there.
- If ever you are running software with graphical displays and the message is something like 'Display not found', run the above commands again. That should fix it.

NOTE! This program needs to run while using Ubuntu to get a display. It's in your Start Menu, and called XLaunch. If you're lazy like me, you can add it to your windows start-up applications:

- Right click on XLaunch in the start-up menu
- Select More → Open file location
- Copy the VcXsrv shortcut file
- Paste the shortcut in `%appdata%/Microsoft/Windows/Start Menu/Programs/Startup`
- (An X icon should appear in your system tray whenever you start Windows)

Step 4/5: Installing required tools on Ubuntu for Windows

For the software used in the course to work, some other packages need to be installed (see blue box below). For easily doing this, run the script on our website. Type the following commands into your Ubuntu terminal: (or copy-paste!)

- `wget http://tbb.bio.uu.nl/BINF/software/Cash2.1/cash_dependencies.sh .` **NOTE:** if you copy this into your terminal window the `_` is often not copied along. Be sure to include it manually, or just download the file `cash_dependencies.sh` from the linked folder in your browser!
- `sudo ./cash_dependencies.sh`
- (optional) If that doesn't work right away, the file might not be marked as an executable. To fix this, type `chmod a+x cash_dependencies.sh`. Then retry the step above and continue.
- (enter your password)
- Test your display by typing `xeyes`. Do you see a window with eyes? If not try the tips under section 3/5 above!

For geeks that want to do it themselves, here's what the script does:

- `apt-get update`
- `apt-get install unzip`
- `apt-get install build-essential`
- `apt-get install grace`
- `apt-get install x11-apps (just for testing)`
- `apt-get install libpng-dev`
- `apt-get install zlib1g-dev`
- `apt-get install libx11-dev`
- `echo "export DISPLAY=localhost:0.0" >> ~/.bashrc`
- `"source ~/.bashrc`

1.3 Installing on Mac OSX

Since Mac OS is also a UNIX system, using it is possible, but it requires some different steps (mostly slightly different dependencies). We need to install a program to manage your display (displaying graphs etc.), like Xquartz. If you are lazy, see if our setup script gets you through it (we've tested the script, but only on a few systems).

To use the setup script, simply do the following:

- `wget http://theory.bio.uu.nl/BINF/software/Cash2.1/cash_dependencies.sh`. **NOTE:** if you copy this into your terminal window the `_` is often not copied along. Be sure to include it manually, or just download the file `cash_dependencies.sh` from the linked folder in your browser!
- `./cash_dependencies.sh` (no `sudo` because brew no longer supports it)
- (optional) If that doesn't work right away, the file might not be marked as an executable. To fix this, type `chmod a+x cash_dependencies.sh`. Then retry the step above and continue.
- (enter your password)
- Test your display by typing `xeyes`. Do you see a window with eyes?

If it didn't work, follow the instructions in the blue box below.

Here's what the script is supposed to do on your Mac OS system. It installs xcode and homebrew, and then uses homebrew to install xquartz, libpng and grace:

- `xcode-select -install`
- `ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`
- `brew install -cask xquartz`
- `brew install libpng`
- `brew install grace`

To edit programs written in a language like C, such as you'll be using in exercise 6, it is preferable to use a code editor. A good choice is VScode. Get it here: <https://code.visualstudio.com/Download>. Alternatives are Notepad++ click, or Atom click. For VScode, be sure to download the

C/C++ extension. You can of course edit the code with whichever application you wish, but these programs help you by highlighting important things, suggesting autocompletions, etc. Be sure to search for handy keyboard shortcuts in VScode or another editor!

1.4 Installing on UU campus (COMPbio students)

Boot the computer in "Betainux" (reboot if it is already in Windows). Open a terminal window (type 'term' in the start menu and it should pop up) and type:

- `/clz/bio-binf/setup-binf`
- `source ~/.bashrc`