
IBiolC Introduction to Bioinformatics Documentation

Release

Leighton Pritchard, Sue Jones, Peter Cock

Mar 12, 2018

This site's contents:

| | | |
|----------|--|----------|
| 1 | What do I need to do before the course? | 3 |
| 1.1 | Installation for Course Attendees | 3 |
| 1.2 | Installation for Tutors | 7 |
| 1.3 | Testing the Installed Tools | 10 |
| 1.4 | Installing Anaconda | 11 |
| 1.5 | Installing ARTEMIS | 12 |
| 1.6 | Installing BLAST | 13 |
| 1.7 | Installing GIT | 14 |
| 1.8 | Installing JALVIEW | 15 |
| 1.9 | Installing JAVA | 15 |
| 1.10 | Installing JMOL | 16 |
| 1.11 | Installing MUSCLE | 17 |
| 1.12 | Installing VirtualBox | 18 |
| 1.13 | Indices and tables | 18 |

This documentation describes software installation and setup for attendees and tutors at the March 2018 presentation of the IBioIC Introduction to Bioinformatics course, for industrial biotechnology postgraduates.

CHAPTER 1

What do I need to do before the course?

You need to ensure that you have the course materials and required software to hand on your own machine.

Attention: COURSE ATTENDEES:

Please follow the installation instructions at *Installation for Course Attendees*.

Attention: COURSE TUTORS:

Please follow the installation instructions at *Installation for Tutors*.

You should have been provided with a static copy of the installation instructions by the course organisers, but online copies of the course materials and a prebuilt virtual machine are also available:

- [Course homepage](#)
- [Course repository](#)
- [Ubuntu Virtual Machine](#)

1.1 Installation for Course Attendees

This document will describe the process for setting up and testing course materials, in preparation for attendance at the workshop.

Two different setups are described:

1. *Installation of all components on your own local machine*
2. *Download and use of a Linux virtual machine*

Either one of these will prepare your machine for the workshop.

We prefer that attendees install materials on their own laptops for the course, as the software and learning materials will then persist and be usable/re-usable on your own machine after the course itself is complete. The virtual machine is a “Plan B” that should be usable on any machine, and is intended as the fallback in case of installation problems on your own laptop.

Important: You should install all the software and/or the virtual machine before you attend the course. This will save you, and everyone else at the course, time at the beginning of the first day, and give more time to explore advanced topics in the workshop.

We will offer an online hangout to help with installation problems, just before the course. Your course organiser will provide information about this.

The process of installing the required software on your machine is as follows:

1. Install `Anaconda` on your computer (if it is not already present).
2. Install `git` on your computer (if it is not already present).
3. Obtain the course materials.
4. Install `Python` requirements.
5. Install third-party software tools.
6. Test the installation.

The process of installing the virtual machine is as follows:

1. Install `VirtualBox` on your computer.
2. Download the course virtual machine and open it in `VirtualBox`.

1.1.1 1. Install all components on your local machine

1. Install `Anaconda` on your computer

For ease of installation and consistency, we install as much as is practical using the `Anaconda` environment. This is cross-platform software that works on Windows, macOS and Linux, and provides several essential components for the course, such as `Python` and the common `Jupyter` notebook interface that will be used.

If you do not already have `Anaconda` installed on your system, please follow the instructions at the page below:

- [*Anaconda installation*](#)

2. Install `git` on your computer

If you do not have a working copy of `git` installed on your machine, install one now by following instructions at the page below.

- [*git installation*](#)

Note: If you are using Windows, you will be installing `git bash` which, as well as `git`, provides the Bash *terminal* environment that you will be using during the course. This provides an experience very similar to working at the terminal in Linux/macOS.

3. Install the course materials

You will need to *clone* the course material repository to your own machine.

Using the terminal¹, navigate to a convenient location (e.g. your Desktop). Then *clone* the course repository with the command:

```
git clone https://github.com/widdowquinn/2018-03-06-ibioic.git
```

Note: This will create a new directory called 2018-03-06-ibioic, containing the course materials.

Change to the course material directory in your terminal with the command:

```
cd 2018-03-06-ibioic
```

4. Install Python requirements

To install the Python module requirements for the course, issue the command below in the terminal¹:

```
pip install -r requirements_students.txt
```

5. Install third-party software

BLAST and MUSCLE do not require JAVA, and can be set up independently:

1. Install BLAST

BLAST is a tool for searching with a protein or nucleotide sequence against a database of other biological sequences.

- [*BLAST installation instructions*](#)

2. Install MUSCLE

MUSCLE is a program for multiple sequence alignment.

- [*MUSCLE installation instructions*](#)

ARTEMIS, JALVIEW and JMOL require the JAVA VM, so JAVA must be installed first:

3. Install JAVA

JAVA is a programming language that runs on a *virtual machine* (the *JVM*). Several bioinformatics tools are written in JAVA, and require the JVM to be installed in order to run.

- [*JAVA installation instructions*](#)

¹ The *terminal* means `git bash` on Windows, and `Bash` on Linux/macOS.

4. Install ARTEMIS

ARTEMIS is a genome sequence browser and editor.

- [*ARTEMIS installation instructions*](#)

5. Install JALVIEW

JALVIEW is a sequence alignment viewer and editor.

- [*JALVIEW installation instructions*](#)

6. Install JMOL

JMOL is a program for visualising biological molecules (e.g. proteins).

- [*JMOL installation instructions*](#)

7. Test the tools/materials

To make sure that the downloaded tools are installed and working on your machine, please follow the instructions on the [*testing your installation*](#) page.

- [*Testing your installation*](#)

1.1.2 2. Download and use a Linux virtual machine

1. Install VirtualBox on your computer

VirtualBox is a program that allows you to run *virtual machines* on your own computer. Virtual machines are software implementations of operating systems that run as if they are a separate computer.

We have provided a virtual machine pre-loaded with software and course materials, as a fallback in case of installation difficulties on your own machine. To install the VirtualBox program, please follow the instructions on the linked page.

- [*Install VirtualBox*](#)

2. Download and import the course virtual machine

We provide a Linux virtual machine pre-loaded with course materials and the required software, which can be used to participate in the workshop.

Attention: The virtual machine file is very large (11GB) and should be downloaded well in advance of the workshop, if you think you may need to use it!

Click on the badge below to go to the virtual machine download page: Click on the link for 2018-03-06-ibioic.vdi to download the virtual machine in a suitable location.

Warning: This may take some time to download!

Click on the link for `2018-03-06-ibioic.vbox` to download the VirtualBox file in the **same location** as the `.vdi` file.

Add the virtual machine.

- Click on Machine in the VirtualBox menu bar
- Click on Add
- Navigate to the `.vbox` file you just downloaded, and select it
- Click on Open

You should see the `2018-03-06-ibioic` machine in the list at the left hand side of the application.

Start the virtual machine.

- Select the new IBioIC virtual machine (`2018-03-06-ibioic`)
- Click on the Start button in VirtualBox

The virtual machine will start as a new window, and appear to be booting up. When this process is complete, it will present you with a login screen. Use the following credentials to log in:

- Username: `ibioic`
- Password: `ibioic-course`

On successful login, you will see a standard Ubuntu desktop, and will be ready to begin the course.

1.2 Installation for Tutors

This document will describe the process for setting up and testing course materials, in preparation for delivery to a class.

Two setups are described:

1. Installation on the local machine
2. Installation on a Linux virtual machine

We prefer that students install materials on their own laptops for the course, as the software and learning materials will then persist and be usable/re-usable after the course itself. The virtual machine approach is a “Plan B” backup option that should be workable on any machine, and is intended as the fallback in case of severe installation problems.

Important: Each new course presentation should be prepared in its own repository. Following the practice of [The Carpentries](#) we have adopted the convention of naming the course repository by date as `YYYY-MM-DD-ibioic`.

The process of installation for course preparation for tutors is as follows:

1. Prepare a repository for your presentation (this may already have been done by a colleague)
2. Create a new [VirtualBox](#) VM for the course
3. Clone the repository to your machine (your laptop, and the VM)
4. Prepare a virtual machine for the course
5. Install required software for the course

6. Test the materials
7. Upload the working VM to ‘[Zenodo](#)’_

1.2.1 1. Prepare a new repository

Note: If one of your colleagues has already created/imported a repository for your presentation, you can skip this part.

When creating a new repository for a new course presentation, please use the [GitHub Importer](#) and provide an existing repository URL to build from, rather than forking an existing repository.

1. Log in to [GitHub](#)
2. In the upper right-hand corner of any page, click + and then `Import Repository`
3. Provide the URL of a previous IBioIC training course repository
4. Choose an account or organisation to own the repository
5. Choose a name for the repository (YYYY-MM-DD-ibioic works for us)
6. Specify that the repository should be *public*
7. Click on `Begin Import`
8. You will receive an email informing you when the repository has been imported
9. Inform the other tutors about the repository and/or invite them as collaborators

1.2.2 2. Create a new VirtualBox VM

If it is not already available, download and install [VirtualBox](#) on your machine. This is a free-to-use general-purpose full virtualizer for x86 hardware, capable of running a virtual machine for use by students on the course.

Tip: The virtual machine for the course may be large, so can be prepared for download by the students well in advance, and include a working *git* installation, so the student can pull an up-to-date copy of the course materials during the course. What is important is that the supporting software are available and can be run on the student’s machine

- [VirtualBox installation](#)

Once [VirtualBox](#) is installed, create a new Ubuntu VM with the same name as your repository for the course presentation.

- Create VirtualBox VM

1.2.3 3. Clone the repository to your machine

Attention: These instructions should be followed to reproduce the repository and required software on both the [VirtualBox](#) VM prepared above, and on your own machine

Warning: If you do not have a working copy of *git* installed on your machine, install one now. This will be required to maintain and publish your repository materials.

- *git installation*

You should clone the repository to your own machine, with the command:

```
git clone <REPOSITORY URL>
```

where *<REPOSITORY URL>* is the repository you have just imported.

Finally, change directory to the root of the new repository.

1.2.4 4. Prepare a conda virtual machine for the course

For ease of installation and consistency, we install as much as is practical using the [Anaconda](#) environment. This is cross-platform on Windows, macOS and Linux, and provides several essential components for the course, such as [Python](#) and the common [Jupyter](#) notebook interface that will be used.

Note: If you do not already have [Anaconda](#) installed on your system, please follow the instructions:

- *Anaconda installation*

Create a new Anaconda environment

With [Anaconda](#) installed, create a new environment with:

```
conda create --name <ENVIRONMENT_NAME> python=3.6
```

Accept all the installation options presented.

Tip: We recommend naming your environment after your repository, using something like 2018-03-06-ibioic as your *<ENVIRONMENT_NAME>*

Then, activate the environment with

```
source activate <ENVIRONMENT_NAME>
```

You should see your terminal prompt change to include the environment name. This is a reminder that you are working within the specific [Anaconda](#) environment for the course materials.

Some tools are useful to us as tutors, for preparing and managing the course materials. These are specified in the file *requirements_tutors.txt*, and should be installed now with the command:

```
pip install -r requirements_tutors.txt
```

1.2.5 5. Install required software for the course

We prefer that students use their own laptops for course delivery, and we aim to match the students' installation experience here, as closely as possible.

Python dependencies

Although we could install most of the python requirements with [Anaconda](#), some of the packages are not available on Windows with this approach, so we install using `pip` instead:

```
pip install -r requirements_students.txt
```

Third-party software

BLAST and MUSCLE do not require JAVA, and can be set up independently:

- *Install BLAST*
- *Install MUSCLE*

ARTEMIS, JALVIEW and JMOL require the JAVA VM, so JAVA must be installed first:

- *Install JAVA*
- *Install ARTEMIS*
- *Install JALVIEW*
- *Install JMOL*

1.2.6 6. Test the materials

1.2.7 7. Upload the VM to ‘Zenodo’_

1.3 Testing the Installed Tools

You would have tested some of these during the installation, but just in case, we’ll recap. Note that the exact version numbers need not match perfectly.

1. Open a terminal window (Git Bash on Windows)
2. Confirm Conda is installed by running:

```
$ conda --version  
conda 4.4.10
```

3. Confirm Git is installed by running:

```
$ git --version  
git version 2.16.2.windows.1
```

4. Confirm Python 3 from Anaconda is installed by running:

```
$ python --version  
Python 3.6.4 :: Anaconda, Inc.
```

5. Confirm the Python libraries we will be using are installed by running:

```
$ python -c "import Bio; import bioservices; import seaborn; import reportlab"
```

6. Confirm Muscle is installed by running:

```
$ muscle -version
MUSCLE v3.8.31 by Robert C. Edgar
```

7. Confirm NCBI BLAST+ is installed by running:

```
$ blastn -v
USAGE
blastn [-h] [-help] ...
```

8. You should have already tested that the Java applications can start.

1.4 Installing Anaconda

1.4.1 Linux Anaconda installation

1. Open <https://www.anaconda.com/download> with your web browser.
2. Download the Python 3 64-bit installer for Linux.
3. Open a terminal window.
4. Type `bash Anaconda3-` and then press `tab`. The name of the file you just downloaded should appear. If it does not, navigate to the folder where you downloaded the file, for example with: `cd ~/Downloads`. Then, try again.
5. Press `[Enter]`. You will follow the text-only prompts. To move through the text, press the `[space]` key. Type `yes` and press `[Enter]` to approve the license. Press `[Enter]` to approve the default location for the files. Type `yes` and press `[Enter]` to prepend Anaconda to your `${PATH}` (this makes the Anaconda distribution the default Python).
6. Close the terminal window.

1.4.2 macOS Anaconda installation

- [Anaconda Video Tutorial \(macOS\)](#)

1. Open <https://www.anaconda.com/download> with your web browser.
2. Download and run the Python 3 installer for OS X.
3. Install Python 3 using all of the defaults for installation.

1.4.3 Windows Anaconda installation

- [Anaconda Video Tutorial \(Windows\)](#)

1. Open <https://www.anaconda.com/download> with your web browser.
2. Download and run the Python 3 installer for Windows.
3. **Install using defaults for installation except**
 - make sure to check **Add Anaconda to my PATH environment variable** (this is required to work with `git bash`)
 - make sure to check **Register Anaconda as my default Python 3.6**.

- you can skip installation of VSCode (though it is a very nice tool)

Warning: You **must** select the Register Anaconda as my default Python 3.6 option on Windows.

1.4.4 Post-installation

We need to add some [Anaconda](#) channels, which is done by issuing the following commands in the terminal¹:

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

Note: On Windows, you can install these channels using the Anaconda Navigator, a graphical tool provided through your Start Menu on that platform.

1.4.5 Starting the Anaconda prompt (Windows-only)

Once Anaconda has been installed, you can start a terminal that “sees” the Anaconda Python installation as follows:

1. Click on the Start/Windows menu
2. Go to Anaconda
3. Scroll down (if necessary) to Anaconda Prompt
4. Click on Anaconda Prompt

This will give you a terminal window where you can run the commands to install Python modules and create conda environments.

Note: We will not be using the Anaconda Prompt as our terminal in this workshop as it does not, by default, understand the Bash commands we will be using to navigate the system.

1.5 Installing ARTEMIS

Warning: ARTEMIS requires JAVA ([installation instructions](#))

We use the genome browser and editor [ARTEMIS](#) at several points in the course. This can be installed following the instructions in the [ARTEMIS manual](#), or as described below:

¹ The *terminal* means either your bash terminal (macOS/Linux), or the git bash terminal (Windows)

1.5.1 macOS installation

The latest version of Artemis is available as a `.dmg` installer:

- [Artemis installer](#)

To install, download the file, uncompress it on your machine, and follow the instructions.

1.5.2 Linux installation

The Linux version of `ARTEMIS` is available as a compressed `.tar.gz` file:

- [Artemis for Linux](#)

This can be downloaded and extracted to produce the `artemis/` directory structure. This can be moved a suitable location (e.g. your home directory) with:

```
mv ./artemis ~/artemis
```

and the `artemis` program added to `${PATH}` with:

```
export PATH=${PATH}:~/artemis/
```

To make this change persist in your system, you should add this line (`export PATH=...`) to your `~/.bash_profile` file.

1.5.3 Windows installation

Download the `ARTEMIS.jar` file:

- [Artemis for Windows](#)

and place it somewhere accessible (e.g. your Desktop).

To start `ARTEMIS`, double-click on the `artemis.jar` file.

1.6 Installing BLAST

We use the sequence search tool `BLAST` at several points in the course. This can be installed as follows:

1.6.1 Linux/macOS installation

`BLAST` is available through `Bioconda`

```
conda install blast
```

1.6.2 Windows installation

1. Download <ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.7.1/ncbi-blast-2.7.1+-win64.exe> with your web browser. If that FTP links does not work, try HTTP instead <http://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.7.1/ncbi-blast-2.7.1+-win64.exe>

2. Run this installer using the defaults, it should put BLAST under `C:\Program Files\NCBI\blast-2.7.1+`
3. Test one of the BLAST programs can be run by executing the command `blastn -h` in the Git Bash terminal.

1.7 Installing GIT

1.7.1 Linux git installation

git should be available in the terminal for your distribution. You can test this by issuing `git --version`:

```
$ git --version
git version 2.15.0
```

If it is not available already, then you should try installing `git` from your distribution's package manager, for example with `sudo apt-get install git` or `sudo apt install git` (Debian/Ubuntu) or `sudo dnf install git` (Fedora).

1.7.2 macOS git installation

git should be available in the terminal. You can test this by issuing `git --version`:

```
$ git --version
git version 2.15.0
```

If it is not available, then you should install the most recent version of the `mavericks` installer from the [Git Mavericks list](#)

1.7.3 Windows git installation

If not installed already, we recommend the use of the `Git Bash` shell throughout the course, for Windows users. This provides a consistent environment equivalent to the powerful Linux and macOS Bash shells.

- [Git Bash video tutorial](#)

1. Download the [Git for Windows installer](#).
2. **Run the installer and follow the steps below:**
 - (a) Click on `Next`.
 - (b) Click on `Next`.
 - (c) Change the editor from default **vim** to use **nano** instead.
 - (d) **Keep “Use Git from the Windows Command Prompt” selected and click on “Next”**. If you forgot to do this programs that you need for the workshop will not work properly. If this happens rerun the installer and select the appropriate option.
 - (e) Click on `Next`.
 - (f) Click on `Next`.
 - (g) **Keep “Checkout Windows-style, commit Unix-style line endings” selected and click on “Next”**.
 - (h) **Keep “Use Windows’ default console window” selected and click on “Next”**.

- (i) Click on `Install`.
- (j) Click on `Finish`.

3. If your **HOME** environment variable is not set (or you don't know what this is):

- (a) Open command prompt (Open `Start` Menu then type `cmd` and press the `[Enter]` key)
- (b) Type the following line into the command prompt window exactly as shown: `setx HOME "%USERPROFILE%"`
- (c) Press `[Enter]`, you should see `SUCCESS: Specified value was saved.`
- (d) Quit command prompt by typing `exit` then pressing `[Enter]`

This will provide you with both `git` and `bash` in the `Git Bash` program, which you can start from the `Start Menu` on your Windows machine.

`git` should be available in the `Git Bash` terminal. You can test this by issuing `git --version`:

```
$ git --version
git version 2.16....
```

1.8 Installing JALVIEW

Warning: JALVIEW requires `JAVA` ([installation instructions](#))

We use the sequence alignment viewer `JALVIEW` at several points in the course. This can be installed as follows:

1.8.1 Linux/macOS installation

`JALVIEW` is available through `Bioconda` (which was set up above):

```
conda install jalview
```

Alternatively, on macOS you can also download the installer `.dmg` file, open it and follow the instructions:

- [JALVIEW installer for macOS](#)

1.8.2 Windows installation

`JALVIEW` can be launched directly from the link below (if `JAVA` is installed):

- [JALVIEW launch](#)

1.9 Installing JAVA

The tools `ARTEMIS`, `JALVIEW`, and `JMOL` are all Java-based, and require the JVM to run. Java is usually present on Linux (and Windows?) machines, but is not available on macOS by default

1.9.1 macOS installation

The latest version of Java is available as a .dmg installer:

- [Java VM installer](#)

To install, download the file, open it and follow the instructions.

1.9.2 Linux installation

Java is likely already installed on your machine. You can test whether it is by issuing the following command at the terminal:

```
java -version
```

This should return a short account of the Java version. If it does not, then please follow the instructions at the page below:

- [Java Installation for Linux](#)

1.9.3 Windows installation

Java is likely already installed on your machine. If it is not, please follow the instructions at the page below:

- [Java Installation for Windows](#)

1.10 Installing Jmol

Warning: Jmol requires JAVA ([installation instructions](#))

We use the protein structure viewer Jmol in the course. This can be installed following the [instructions](#) on the Jmol website.

1.10.1 Download Jmol

Jmol is provided as a single JAVA application for all operating systems. To download it, click on the link below.

- [Jmol latest version](#)

Clicking on the link above should download a .zip or .tar.gz file, which can be extracted, producing a directory containing the Jmol application.

1.10.2 Running Jmol

To start Jmol, open the parent directory in your file explorer (e.g. Finder on macOS), and double-click on the jmol.jar file.

Note: On macOS you may not be permitted to run this executable, as the program is not signed. If this is the case, open System Preferences -> Security & Privacy and click on the General tab. In the lower section

of the window, you should see an option to trust the file `jmol.jar`. Accept this offer. As soon as you do this, Jmol should start.

Alternatively, you can start Jmol from the command-line. Navigate to the directory containing `jmol.jar` and issue the following command:

```
java -jar jmol.jar
```

The splash screen should appear, and the application should start.

1.11 Installing MUSCLE

We use the sequence alignment tool **MUSCLE** in one section of the course. This can be installed as follows:

1.11.1 Linux/macOS installation

MUSCLE is available through **Bioconda** (which should already be set up on your machine):

```
conda install muscle
```

1.11.2 Windows installation

At the time of writing, **MUSCLE** is not available through **Bioconda** for Windows.

1. Open <http://drive5.com/muscle/downloads.htm> with your web browser.
2. Download the latest Windows Intel i86 binary, currently `muscle3.8.31_i86win32.exe`. This will be placed in your Downloads directory.
3. In `git bash` change to your home directory with the command `cd`.
4. Create a new directory called `bin` with the command `mkdir bin`.
5. Copy the **MUSCLE** program to this new directory with the command `cp Downloads/muscle3.8.31_i86win32.exe bin/muscle.exe`. This creates a new command called `muscle` which runs the alignment program.
6. Test that the program can be run by executing the command `muscle` in the terminal.

In total, the sequence of commands will be:

```
$ cd
$ mkdir bin
$ cp Downloads/muscle3.8.31_i86win32.exe bin/muscle.exe
$ muscle
```

Note: The `$` sign should not be typed - this indicates the command prompt you will see on your screen.

1.12 Installing VirtualBox

VirtualBox is a free-to-use general-purpose full virtualizer for x86 hardware, capable of running a virtual machine for use by students on the course.

- [Download page](#)

Download the appropriate binaries for your system, and follow the installation instructions.

1.12.1 macOS installation

The installer downloads as a .dmg file.

1. Double-click on the .dmg file to open it. This will open a new `Finder` window showing the file contents.
2. **Double-click on the `VirtualBox.pkg` package, and follow the instructions**
 - Click `Continue` to run the package to determine if the software can be installed
 - Click `Continue`
 - Click `Install` to select the standard installation location (you will be prompted for your password)
 - The installation should report success. Click `Close` to end.

1.12.2 Linux installation

Follow the instructions for your distribution at the page below:

- [Installing VirtualBox on Linux](#)

1.12.3 Windows installation

Download and run the installer package from the page below:

- [VirtualBox downloads](#)

1.13 Indices and tables

- [genindex](#)
- [search](#)