Python for Biologists - Local Development Setup

Windows Installation Guide

Course: Year 3 Biology Python Programming **Instructor**: Helfrid Hochegger (hh65@sussex.ac.uk) **Last Updated**: 2025-01-14

Overview

This guide will walk you through setting up a professional Python development environment on your Windows machine. Follow these steps in order!

What you'll install:

- ✅ Warp Terminal (modern terminal with AI features)
- ✅ Git for Windows (for downloading course materials)
- ✅ UV (Python package manager)
- ✅ Cursor IDE (code editor with AI assistance)
- ✅ Course notebooks repository

Estimated Time: 30-45 minutes

Step 1: Install Warp Terminal

Warp is a modern terminal with AI features that makes command-line work much easier for beginners.

Windows Users

1. Visit warp.dev
2. Click "Download for Windows"
3. Run the downloaded installer (.exe file)
4. Follow the installation wizard
5. Launch Warp
6. Sign in with your email (free account)

Alternative: You can use PowerShell or Windows Terminal if you prefer. All commands work the same!

Step 2: Install Git for Windows

Git for Windows includes Git Bash, which gives you a Unix-like terminal on Windows.

Download and Install:

1. Visit git-scm.com/download/win
2. Download the installer (64-bit recommended)
3. Run the installer

Important Installation Options:

When the installer asks, choose these settings:

- ✅ **Adjusting your PATH**: "Git from the command line and also from 3rd-party software"
- ✅ **Choosing the SSH executable**: "Use bundled OpenSSH"
- ✅ **Choosing HTTPS transport**: "Use the OpenSSL library"
- ✅ **Line ending conversions**: "Checkout Windows-style, commit Unix-style"
- ✅ **Terminal emulator**: "Use MinTTY (default terminal of MSYS2)"
- ✅ Everything else: use defaults

Verify Installation:

Open Warp (or PowerShell) and run:

```
git --version
```

You should see something like `git version 2.43.0.windows.1`

✅ **Success!** Git is installed.

Step 3: Clone the Course Repository

Now let's download all the course materials from GitHub.

Open Warp or Git Bash

For the rest of this guide, we'll use Git Bash syntax (works in Warp too).

Navigate to your Documents folder:

```
cd ~/Documents
mkdir biology-python
cd biology-python
```

Note: ~ means your home folder (like `C:\Users\YourName\`)

Clone the repository:

```
git clone https://github.com/HocheggerLab/y3-bio-python.git
```

You'll see output like:

```
Cloning into 'y3-bio-python'...
remote: Enumerating objects...
Receiving objects: 100% done.
```

Navigate into the repository:

```
cd y3-bio-python
```

✅ **Success!** You now have all the course notebooks on your computer.

Step 4: Install UV (Python Package Manager)

UV is a modern, fast Python package manager that will handle all your dependencies.

Install UV:

Open **PowerShell as Administrator**:

1. Press Windows key
2. Type "PowerShell"
3. Right-click "Windows PowerShell"
4. Select "Run as administrator"

Then run:

```
powershell -c "irm https://astral.sh/uv/install.ps1 | iex"
```

Alternative for Git Bash/Warp:

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

Restart your terminal

Close and reopen Warp (or your terminal).

Verify installation:

```
uv --version
```

You should see the UV version number!

Troubleshooting: If you get “uv: command not found”:

1. Close and reopen your terminal
2. Or manually add to PATH: `C:\Users\YourName\.cargo\bin`

Step 5: Set Up Python Environment

Now let’s install all the course packages (numpy, pandas, matplotlib, jupyter, etc.)

Make sure you’re in the course repository folder:

```
cd ~/Documents/biology-python/y3-bio-python
```

PowerShell users: Use this instead:

```
cd $HOME\Documents\biology-python\y3-bio-python
```

Sync all course packages:

```
uv sync
```

This will:

- Read the `pyproject.toml` file
- Create a virtual environment (`.venv` folder)
- Install Python 3.12 and all course packages

This takes 2-3 minutes. You’ll see output like:

```
Resolved 23 packages in 450ms
Installed 23 packages in 2.1s
```

✅ **Done!** Your Python environment is ready.

Step 6: Download and Install Cursor IDE

Cursor is like VS Code but with AI superpowers - perfect for learning Python!

Download Cursor:

1. Visit cursor.com
2. Click “Download for Windows”
3. Run the installer (.exe file)
4. Follow the installation wizard

Launch Cursor:

1. Open Cursor from Start Menu or Desktop shortcut
2. Grant any permissions if prompted (Windows Firewall, etc.)

Sign in (optional but recommended):

- Click “Sign In” in the bottom-left
- Create a free account or sign in with GitHub
- This enables AI features

Step 7: Open the Repository in Cursor

Option 1: From Terminal

In your terminal (make sure you’re in the `y3-bio-python` folder):

```
cursor .
```

This opens the current folder in Cursor!

Option 2: From Cursor

1. Open Cursor
2. File → Open Folder
3. Navigate to `C:\Users\YourName\Documents\biology-python\y3-bio-python`
4. Click “Select Folder”

Step 8: Install Required Cursor Extensions

You need two extensions to work with Jupyter notebooks.

Install Python Extension:

1. Click the Extensions icon in the left sidebar (looks like building blocks)
   - Or press `Ctrl+Shift+X`
2. Search for “Python”
3. Install the one by **Microsoft** (it has the most downloads)

Install Jupyter Extension:

1. In the Extensions sidebar, search for “Jupyter”
2. Install the one by **Microsoft**
3. **Restart Cursor** after installation

Note: Cursor usually prompts you to install these automatically when you open a .ipynb file!

Step 9: Select Python Interpreter

Tell Cursor to use your virtual environment.

Steps:

1. Press `Ctrl+Shift+P` to open the command palette
2. Type "Python: Select Interpreter"
3. Select the interpreter that shows `.venv` in the path
    - Should look like: `Python 3.12.x ('.venv': venv)`
    - Path might show: `.venv\Scripts\python.exe`

You only need to do this once per project!

Step 10: Test Your Setup

Let's make sure everything works by running a notebook!

Open a notebook:

1. In the Cursor file explorer (left sidebar), navigate to:

   `notebooks/lecture_1/`

2. Click on `L1_N1_collab_notebooks.ipynb`

Run the first cell:

1. Look for the first code cell in the notebook
2. Click the ▶️ play button next to it
    - Or press `Shift+Enter`
3. Wait a moment for the kernel to start

What should happen:

- The cell runs
- You see output below the cell
- A number appears in brackets like `[1]` next to the cell

If it asks you to select a kernel:

1. Click "Select Kernel" at the top-right of the notebook
2. Choose "Python Environments…"
3. Select the one with `.venv` in the name

✅ You're All Set!

Congratulations! Your development environment is fully set up. You can now:

- ✅ Open any notebook in the `notebooks/` folder
- ✅ Run cells with `Shift+Enter`
- ✅ Ask Cursor AI for help (press `Ctrl+L`)
- ✅ Work offline with all course materials

Quick Reference Commands

Git Bash / Warp:

```
# Navigate to your project
cd ~/Documents/biology-python/y3-bio-python

# Open in Cursor
cursor .

# Get latest course updates
git pull

# Add new Python packages (if needed)
uv add package-name
```

PowerShell:

```
# Navigate to your project
cd $HOME\Documents\biology-python\y3-bio-python

# Open in Cursor
cursor .

# Get latest course updates
git pull

# Add new Python packages (if needed)
uv add package-name
```

Troubleshooting

"git: command not found"

- Make sure Git for Windows is installed (Step 2)
- Restart your terminal
- Check if Git Bash is available in your Start Menu

“uv: command not found”

- Restart your terminal after installing UV
- Run PowerShell as Administrator and reinstall
- Manually add to PATH: Settings → System → Environment Variables → Edit “Path” → Add `C:\Users\YourName\.cargo\bin`

PowerShell execution policy errors

Run PowerShell as Administrator:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope
        CurrentUser
```

Notebook won’t run / “No kernel”

- Make sure you selected the `.venv` interpreter (Step 9)
- Try restarting Cursor
- Make sure both Python and Jupyter extensions are installed

“Cannot find module ‘numpy’” or similar

- Make sure you ran `uv sync` in Step 5
- Make sure you selected the correct interpreter (should show `.venv`)
- The path should show `.venv\Scripts\python.exe`

Extension installation fails

- Check your internet connection
- Check Windows Firewall isn’t blocking Cursor
- Try restarting Cursor and installing again

Path with spaces causes errors

Always use quotes around paths with spaces:

```
cd "C:\Users\Your Name\Documents\biology-python"
```

Windows Defender slows down installation

If installations are very slow:

1. Temporarily disable real-time scanning
2. Add exceptions for:
    - `C:\Users\YourName\.cargo\`
    - Your project folder
    - Cursor installation folder
3. Re-enable after installation

Windows-Specific Tips

File Explorer Integration

To open a folder in Cursor quickly:

1. Navigate to the folder in File Explorer
2. Right-click inside the folder
3. Look for "Open with Cursor" (if configured during install)

Using WSL (Advanced Alternative)

If you're comfortable with Linux and want a Unix environment:

1. Install WSL2: Open PowerShell as Admin and run `wsl --install`
2. Follow the Mac/Linux guide instead
3. Access files via `\\wsl$\` in File Explorer

Windows Terminal (Alternative to Warp)

If Warp doesn't work or you prefer Microsoft's terminal:

1. Install from Microsoft Store: "Windows Terminal"
2. It can run PowerShell, Git Bash, and more in tabs
3. All commands in this guide work the same

Getting Help

- **Cursor AI**: Press `Ctrl+L` and ask questions about your code
- **Course Seminars**: Bring setup questions to practical sessions
- **Course Forums**: Post questions - other students can help!
- **Instructor**: Email hh65@sussex.ac.uk for technical issues

---

What's Next?

1. **Explore the notebooks**: Start with `notebooks/lecture_1/`
2. **Practice running cells**: Get comfortable with the interface
3. **Try Cursor AI**: Press `Ctrl+L` and ask it to explain Python concepts
4. **Attend seminars**: We'll work through exercises together!

Remember: Focus on learning Python, not mastering the IDE. The tools are here to help you code!

Happy coding! 🐍🔬