Sambuca/Bioopti on Windows

Contents

[Install Git 1](#_Toc438021771)

[Optional - Git GUI 1](#_Toc438021772)

[Create a Sambuca\_Project Directory 1](#_Toc438021773)

[Install WinPython 2](#_Toc438021774)

[Fix for IPython/Jupyter Notebooks on CSIRO Windows Systems 2](#_Toc438021775)

[Clone the Code Repositories 2](#_Toc438021776)

[Install The Packages 4](#_Toc438021777)

[Test your Installation 4](#_Toc438021778)

[Install and Configure the PyCharm IDE 5](#_Toc438021779)

# Install Git

<http://git-scm.com/download/win>

During the install, keep all the default options (safest).

For more info on using Git, see one of the many online Git tutorials. A good reference is <https://www.atlassian.com/git/tutorials/>

Scientific Computing also have a [Git Learning Station](https://wiki.csiro.au/display/ASC/7+Version+Control) that will help with learning the basic commands.

## Optional - Git GUI

If you want a Windows GUI for working with Git, the best I have found is SourceTree:

<http://www.sourcetreeapp.com/>

# Create a sambuca\_project Directory

Create a directory called “sambuca\_project” where you wish to keep all your Sambuca and BioOpti files. You will need at least 2GB of free space due to the WinPython distribution.

# Install WinPython

[WinPython](http://winpython.github.io/) is a portable scientific Python distribution.

1. Download one of the Python 3.4 or 3.5 [WinPython installers from SourceForge](http://sourceforge.net/projects/winpython/files/).
2. Run the installer and select your sambuca\_project directory.

# Fix for IPython/Jupyter Notebooks on CSIRO Windows Systems

If you have tried running IPython Notebooks on your CSIRO Windows PC, you may have noticed that it simply doesn't work. Instead the browser stops responding as soon as you launch a notebook. The problem is a port conflict between the VNC Server instance used for remote administration of Windows PCs and one of the libraries used by IPython Notebooks.

Until a fix works its way through the open source pipeline, the IM&T Desktops team have developed a workaround for this issue. Simply run the following script to change the VNC Server port on your PC:  
[\\ds.csiro.au\resources\Scripts\IPython-VNCport-fix\IPythonNotebookFix.bat](file:///\\ds.csiro.au\resources\Scripts\IPython-VNCport-fix\IPythonNotebookFix.bat)

This adjusts the port used by VNC Server so that it no longer conflicts with the IPython Notebook server.

You only need to run this script once. If you have already run it in the past, then running it again is still safe.

# Clone the Code Repositories

The required repositories are all contained in a project on CSIRO’s Bitbucket server: <https://bitbucket.csiro.au/projects/SAM>

Running Bioopti and Sambuca requires the [bioopti](https://bitbucket.csiro.au/projects/SAM/repos/bioopti/browse), [bioopti\_data](https://bitbucket.csiro.au/projects/SAM/repos/bioopti_data/browse), [sambuca](https://bitbucket.csiro.au/projects/SAM/repos/sambuca/browse), and [sambuca\_core](https://bitbucket.csiro.au/projects/SAM/repos/sambuca_core/browse) repositories.

Regardless of whether you use SourceTree or the Git command line, clone all repositories to the sambuca\_project directory you created earlier.

There are two ways of cloning the repositories:

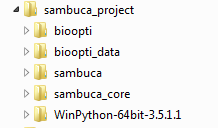
The easiest is to use SourceTree:

* 1. Install SourceTree if you haven't already done so.
  2. Point your browser at the required repository.
  3. Click the "…" button at the top left of the page, and select "Clone" from the list of options.
  4. Select the "Clone in SourceTree" button
  5. Follow the SourceTree wizards to complete the process.

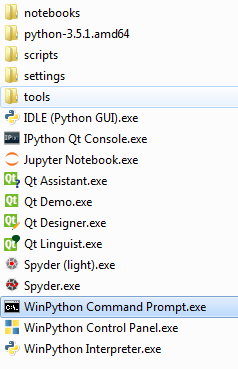
To clone with the Git command line:

* 1. In Windows File Explorer, select the directory that you want to be the parent to the Sambuca code.
  2. Right click inside the directory and select "Git Bash Here" from the context menu.
  3. Point your browser at the required repository.
  4. Click the "…" button at the top left of the page, and select "Clone" from the list of options.
  5. Copy the HTTP URL to the clip board (control-c)
  6. At the Git command line, type "git clone " and then shift-insert to paste the repository URL into the command line.
  7. Press enter.

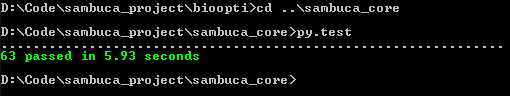
Your sambuca\_project directory should now look something like this:



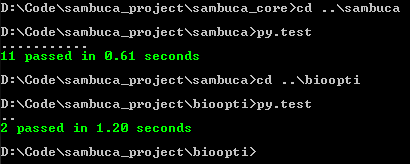
# Install The Packages

* 1. In the WinPython directory, execute “WinPython Command Prompt.exe”  
     
  2. Change up 2 directories to get back to sambuca\_project:  
     cd ..\..
  3. cd to sambuca\_core and execute the following command:  
     python setup.py develop
  4. Repeat the “python setup.py develop” command for sambuca and bioopti (cd to the directory first)

# Test your Installation

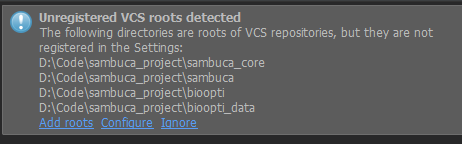
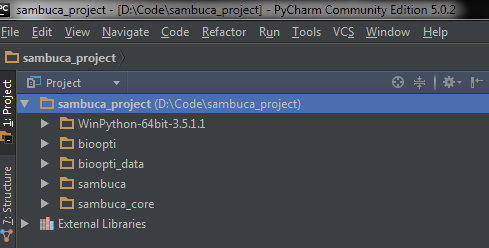
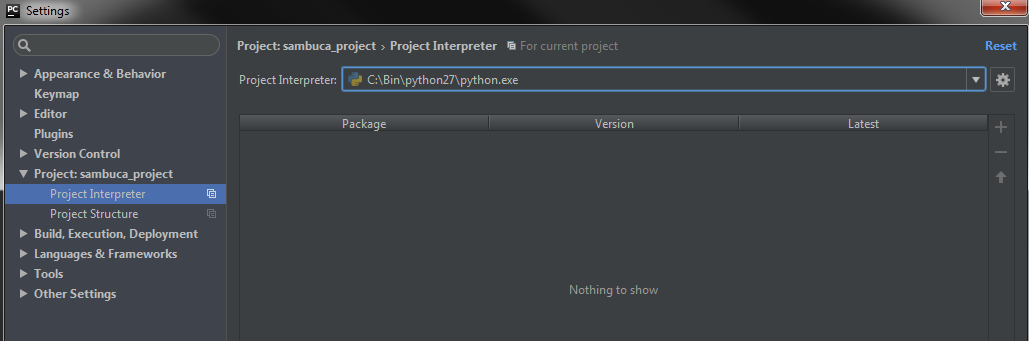
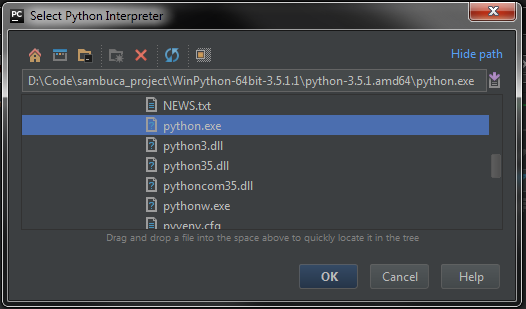
Still in the WinPython command prompt, change to the sambuca\_project\sambuca\_core directory and type ‘py.test’ to run the unit tests:  


You can also run the sambuca and bioopti unit tests:



# Install and Configure the PyCharm IDE

The best IDE to use on Windows is [PyCharm Community edition](https://www.jetbrains.com/pycharm/). If you have Visual Studio, then [Python Tools for Visual Studio](http://microsoft.github.io/PTVS/) can also be useful, but I haven’t tested it with this project.

1. Download and install PyCharm Community edition now (unless you already have it).
2. Run PyCharm and open the sambuca\_project directory as a project in PyCharm.
3. You should see a warning about unregistered VCS roots:  
     
   This is normal. PyCharm has built in Git support and can be used instead of the Git command line or SourceTree. The best option here is to select the Add Roots option to add the source control information to your PyCharm project.
4. You should now see something like this in the PyCharm project view:  
   
5. We now need to configure PyCharm to use the WinPython interpreter for the project. From the PyCharm File menu, select “Settings”, and then the project interpreter settings panel:  
     
     
   Typically it won’t find the WinPython installation, so using the “Project Interpreter” drop-down list, select “show all” to bring up the Project Interpreters window. Click the green plus button to add a new local interpreter and select the WinPython executable:  
   
6. Click OK on all the windows to get back to the main PyCharm GUI.
7. Add PyCharm configurations for running the unit tests. In the default GUI, at the top-right toolbar you will see a small arrow pointing down:  
     
   Click it and select “edit configurations”
   1. Click the green plus sign to add a new configuration,