## Requirements

* A working QGIS installation (2.4 or greater) with the Processing toolbox installed and working. If Processing is installed, a menu item named ‘Processing’ should exist in QGIS
* An updated QGIS Python installation. QGIS Python comes with old versions of numpy, scipy, and shapely. Updating those packages can be done manually by overwriting the appropriate folders in site\_packages with new ones, or using pip as described [Here](https://woostuff.wordpress.com/2012/12/19/installing-python-setuptools-into-osgeo4w-python/). The versions required are,
  + Numpy 1.7.1 or above
  + SciPy 0.13 or above
  + Shapely 1.5.1 or above
* Matplotlib 1.2.1 or above and all dependencies.
* Tkinter (only if not included with QGIS Python). Can extract it from the standard Python installation. On Windows, need the following files
  + Into the apps\Python27\Lib folder, copy the entire lib-tk folder from the standard Python installation.
  + Into the apps\Python27 folder, copy the entire libs folder (don’t overwrite any files already in that folder).
  + Into the apps\Python27\DLLs folder, copy \_tkiniter, tcl85.dll, tclpip85.dll, tk85.dll
  + Into the apps\Python27 folder, copy the entire tcl folder.
* The QGIS ScriptRunner plugin. Use the plugin manager (Plugins -> Manage and Install Plugins) to install it.
* A working Wiggly installation.

This example was tested on Window 7 64 bit using the above (32 bit) versions of the software.

## Introduction

This problem shows how to use Wiggly to analyze a Python model written for the Processing toolbox. The file ‘buffer\_area\_dist.py’ is a Processing script compatible with the Wiggly-QGIS adapter. This script calculates the area of a point, buffered out to a user-defined distance.

In order to conduct an analysis with Wiggly, two components are needed: a control script and the model to analyze. This is similar to using Wiggly outside of QGIS. The difference is that the control script and model scripts need to have some extra code to allow the adapter to correctly link them with Wiggly’s engine. In this example,

* buffer\_area\_dist.py = the analysis model
* UQ\_buffer\_area\_dist.py = the control script

This document only goes through how to run the example. See the code in the above two files for details on how analyze your own models.

## Running the Example

The ‘buffer\_area\_dist.py’ is a script written for QGIS’s Processing toolbox. Load it into the toolbox by clicking ‘Add script from file’ (Figure 1). If successful, the script should appear under the Scripts 🡪 User scripts branch of the Processing Toolbox.

Test the script by double-clicking it from the processing toolbox. Press the run button and you should get a point layer appearing in QGIS. In the output log you should get the output ‘Total area: 3.14146346236’ among other things. If the output closes too fast, change the setting in the Processing toolbox that leaves the output window open after the script runs.

Now, open the ScriptRunner plugin (Figure 2) and add the control script by clicking on the green ‘+’ icon on the toolbar. If successful, the control script should appear in the list of scripts (Figure 3).

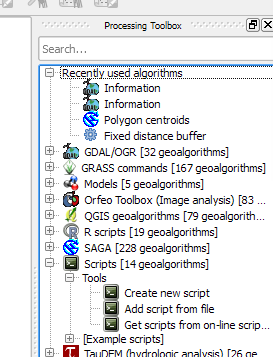


Figure 1

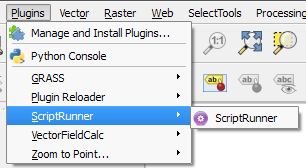


Figure 2

Now, click the green arrow in ScriptRunner to run the uncertainty quantification analysis. After a few moments, the results (Figure 4) should appear. If you get a ‘Bad File Descritptor’ error message try running again (this is a QGIS bug).

The files containing the run are saved in ‘c:\temp’. This location may be changed from the control script.

## Troubleshooting

If adding the UQ\_buffer\_area\_dist.py script to ScriptRunner generates an error, it usually means that Wiggly, the Wiggly-QGIS adapter or one of its dependencies could not be initialized. To get more information, open the QGIS Python console and try importing those modules manually.

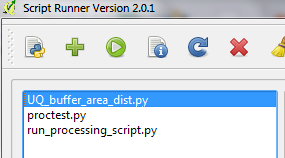


Figure 3

Occasionally, runs will fail with the error 'IOError: [Errno 9] Bad file descriptor' in the text. This is a problem with QGIS. Re-running the analysis will normally resolve the problem. Note that if this error appears at the end of a run, the model results should still be usable, though they will require manual plotting with either Wiggly’s plot functions or directly with matplotlib.

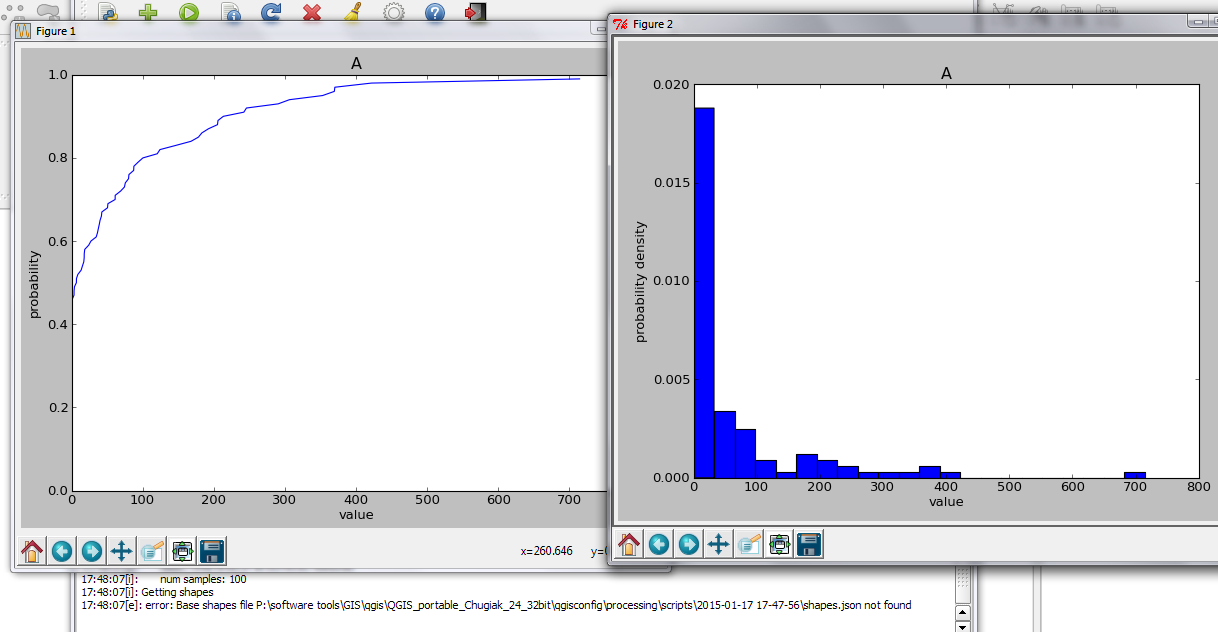


Figure 4: example results.