Quantum Programming Without Tears

Max Cutugno Josh Gordon Tino Tamon

June 8, 2018

Contents

Chapter 1

Python based platforms

Pyquil and QISkit are based on python. In what follows, we describe some preliminary setup options for running the quantum python programs for these two platforms.

In what follows, we describe an installation on Ubuntu 14.04 running on a 64-bit machine. The installations on Windows 10 and Mac OS X are similar (and somewhat easier).

1.1 Anaconda

To install python in a self-contained environment (which may avoid subtle clashes between python2 and python3 as well as their respective libraries), one option is to use anaconda. We stress that this is optional as one can install pyquil directory without using anaconda.

First, we install anaconda into some directory, say `anaconda2`. Then, to make it visible globally, we update the bash file `.bashrc`:

```
#Place the following line in the file .bashrc (at the end)
export PATH="$USER/anaconda2/bin:$PATH"
```

Here the variable `$USER` specifies the path to the user's main directory. Make sure to source the file from the command line:

```
$ source .bashrc
```

Now conda should be visible from anywhere.

We can update conda first (to catch any latest upgrades):

```
$ conda update conda
```

Now, we create a conda *environment* for running our python based programs:

```
# create a new conda environment called 'myquil'
$ conda create --name myquil python=python3.6
$ conda env list
```

To enter the conda environment, we do

```
$ source activate myquil
(myquil) $
```

and to exit we do

```
(myquil) $ source deactivate
$
```

1.2 Jupyter Notebook

If one wishes to use Jupyter notebook, we need to the add the conda environment to the notebook. First, we enter the conda environment and then install a python kernel for jupyter notebook and connect it to the conda environment:

```
(myquil) $ pip install ipykernel
(myquil) $ python -m ipykernel install --user --name myquil
```

Now, we can run jupyter notebook (from outside conda) and choose the appropriate kernel (and hence conda environment).

Chapter 2

Pyquil

On Linux, we can install pyquil directly (using pip). The installations on Windows 10 and MacOS X are similar (we might need brew on MacOS X instead of pip).

2.1 Anaconda

But, we can also install it within a conda environment as follows:

```
$ source activate myquil
(myquil) $ pip install pyquil
(myquil) $ pyquil-config-setup
Welcome to PyQuil!
Enter the required information below for Forest connections.
If you haven't signed up yet you will need to do so first
  at https://forest.rigetti.com
Forest API Key: <ENTER API HERE>
User ID: <ENTER ID HERE>
Pyquil config file created at '/home/tino/.pyquil_config'
If you experience any problems see the guide
  at https://go.rigetti.com/getting-started
```

Now, we can run some python programs in a directory of examples:

```
(myquil) $ python run_quil.py hello_world.quil
Running Quil Program from:  hello_world.quil
---------------------------
```

```
Output:
[[1, 0, 0, 0, 0, 0, 0, 0]]
```

Chapter 3

QISkit

Chapter 4

Quipper

This seems to be the hardest to install. QLSA purportedly is implemented here.

Chapter 5

Liquid

This seems to be the most straightforward to install. It is unclear if QLSA runs correctly here.

Chapter 6

Sanity Checks

6.1 Main algorithms

We examine several basic known quantum algorithms and verify if they are implemented within a certain quantum software platform.

NAME	Pyquil	QISkit	Quipper	Liquid
Bell circuit	✓	✓	✓	✓
teleportation	✓	✓	✓	✓
Deutsch	×	×	×	×
Deutsch-Jozsa	×	×	×	×
Bernstein-Vazirani	×	×	×	×
Simon	×	×	×	×
QFT	×	×	✓	✓
Phase estimation	×	×	×	×
Shor	×	×	×	×
Grover	×	×	×	×
Singular value estimation	×	×	×	×
Hamiltonian simulation	×	×	×	×
HHL	×	×	×	×

6.2 Additional algorithms

1. (Dürr and Høyer) Finding the minimum element in an array.

2. (Ambainis) Determining if an array contains distinct items.

3. (Child and Goldstone) Continuous-time spatial search.

Chapter 7

Graphical quantum circuit tools

Chapter 8

Cross compilation

Chapter 9

Baffling questions

1. Why does Quipper provide a *cloning* function?
2. In Liquid, is F# now obsolete and being replaced by Q#? If so, why?