Azure in-house PCR analysis software, azureIPA

Ryan Koehler
ryan@verdascend.com
8/17/19

Background.

Real-time PCR yields data consisting of fluorescence values collected at subsequent reaction cycle numbers. Ideally, fluorescence is directly proportional to the number of amplified DNA molecules, and this theoretically doubles at each cycle. From such a “curve” one determines a Cq value, which is the value on the cycle number axis where the florescence rises above a background signal threshold, and from this can calculate the starting concentration of target DNA in the starting sample. There are numerous methods to choose thresholds, do “baseline correction,” calculate Cq values, and calculate sample DNA concentrations.

PCR reactions are typically run in 96-well plates (i.e. 96 separate reactions), and single reaction wells may contain several different assays that each yield different florescence signals. The different florescence signals are captured in different optical channels. Thus, raw PCR data may consists of florescence values for multiple channels for each of 96 wells over the course of multiple reaction cycles (typically 40 cycles).

In addition to running PCR reactions to identify Cq values, a typical experiment often generates product “melting” data which is used to determine characteristics of the PCR products. This data consists of florescence values sampled during a ramp of temperatures (i.e. low to high). PCR product DNA (“amplicons”) will exist as double stranded molecules at low temperatures but then separate into single stranded DNA at higher temperatures. At what temperature this occurs, as well as the sharpness of the transition, can be informative (i.e. the reaction products look “good” or suspicious).

Current status.

The Azure in-house PCR analysis software, *azureIPA*, is intended to allow interactive exploration of rt-PCR and melting data. The program consists of a GUI which has two panels to display fluorescence data curves, a grid emulating the layout of a 96-well plate to allow selection of specific wells, and a text window to report Ct values. Data with multiple channels per well are supported, allowing different channels to be plotted in different colors (or hidden). Plots can display raw data, base-corrected data, and first or second derivatives of curves. Plots also support auto-range or custom range zoom (i.e. florescence signal and/or cycle number). Fluorescence (Y axis) may be set to log scale. Thresholds are automatically determined (very basic algorithm!) but these may also be manually adjusted (independently for each channel present in the data).

Current *azureIPA* program is primarily a GUI framework. Various algorithms for PCR / melting analysis could be implemented and explored, but the scientific content of current code has only rudimentary functionality here. In addition, the GUI includes some menu items (“features”) that have not been implemented (like simulations) but could be useful.

When loading and playing with large datasets (96 wells, multiple channels), program response time can be slow. Only minimal efforts to increase graphic efficiency (e.g. by limiting redraws) have been attempted. Redrawing in response to user interaction is somewhat problematic, especially for selection / deselection of wells within the “Plate view” window. For example, when wells are selected / deselected, updates to graphic plots and text reports are not always forthcoming. Improvements to GUI performance would be most welcome.

Software architecture, dependencies.

Code was developed as Python 3.7 on Ubuntu Linux 16.04. The GUI uses the *wxPython* library. Data is held and processed using the *Pandas* library. Plotting is done via *matplotlib* and both *Pandas* and matplotlib require *numpy*. All of these libraries are required to run the program. Thus, Python 3 and the two mentioned libraries need to be installed in order to run *azureIPA*.

The code has also been tested on Windows; Some GUI elements appear slightly different (as expected for a different windowing system), but the core functionality (i.e. plotting, setting thresholds, etc) appears to work fine. In addition to running the native python code, a Windows executable may be created via the *py2exe* tool. The resulting exe files are huge but they do work (and only the single huge file is needed to run on the Windows machine).

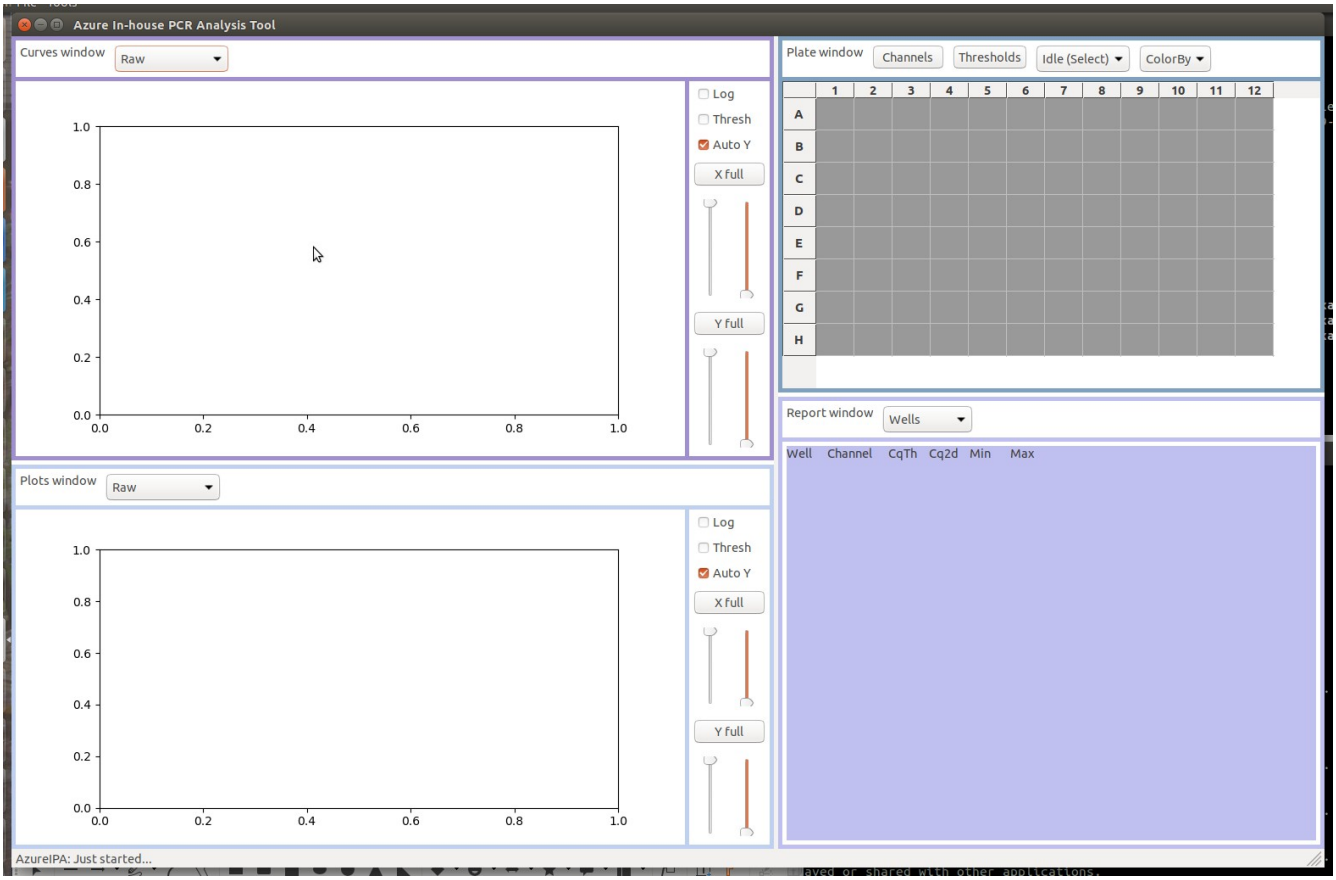
Example use.

Several example data files are included with the code. These are simple text (csv) files; Lines starting with “#” are comments and ignored. Filenames indicate the number of different channels (i.e. 1, 3, or 6) and the data type; “_rt” and “_mc” denote real-time PCR data and melting data, respectively. Thus there are three rt-PCR files and a single melting data file. Data in the “azexam3_*” files are in three channels but no wells have more than a single channel of data. Data in the “azexam6_rt.csv” file are in six channels, with multiple channels per well. However, some channel data are not meaningful for PCR analysis (i.e. they are essentially background noise data, etc).

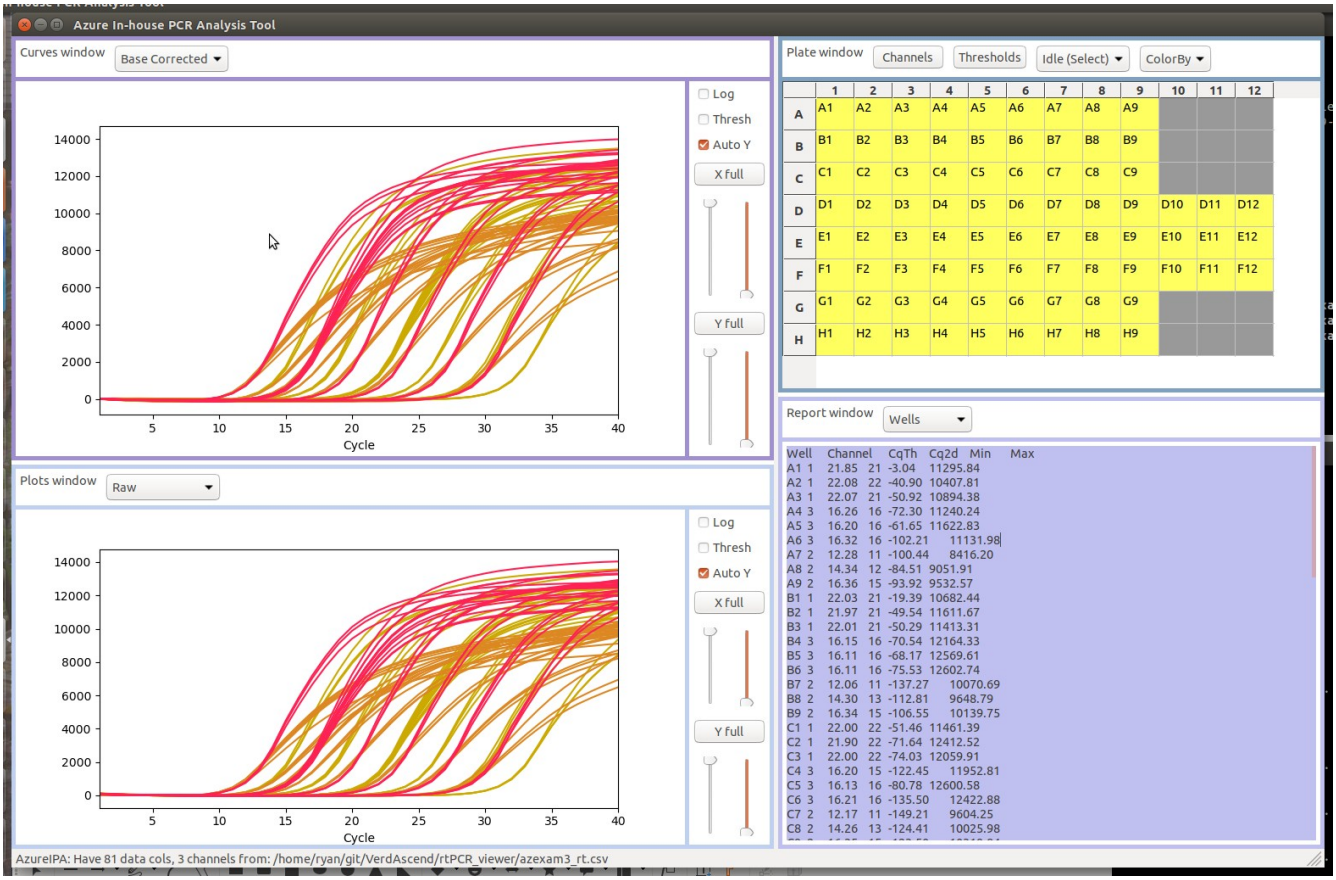
```
azexam1_rt.csv
azexam3_mc.csv
azexam3_rt.csv
azexam6_rt.csv
```

The below steps and screenshots illustrate how one can explore data with *azureIPA*.

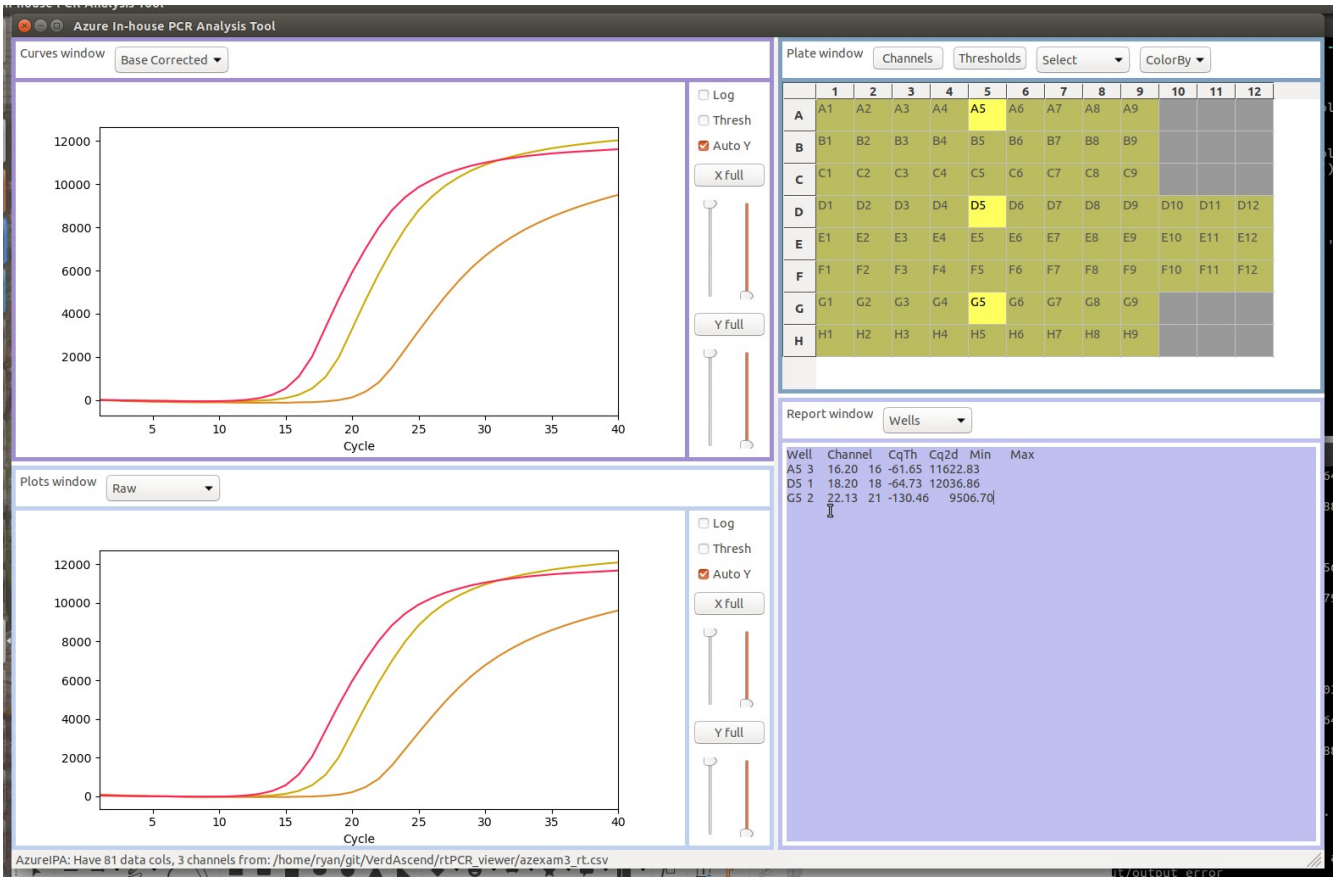
1) Launch *azureIPA*. (command line; simply call by name).



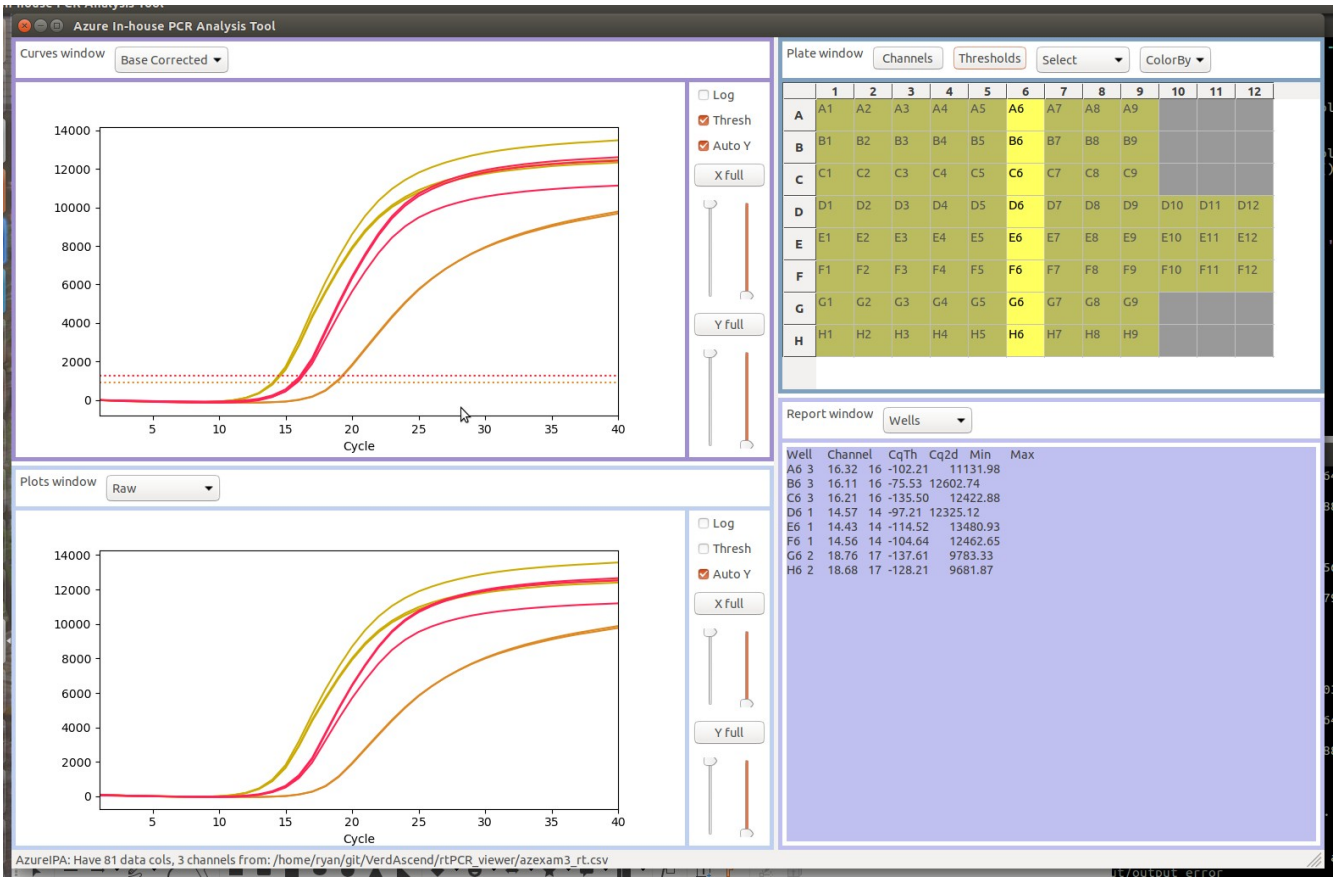
2) Load the rt-PCR data. This is done via the “file” pull-down menu: Open > Data then specify “`azexam3_rt.csv`”. Upon loading the data, all channels (three in this case) are displayed, with the raw data shown in the lower left panel (“Plots window”) and the baseline-corrected data shown in the upper left (“Curves window”); Baseline correction simply shifts values so that the first value (cycle 1) is set to zero. The upper right (“Plate window”) indicates 96-well plate wells that have data; Grey cells have no data, and bright yellow indicates that all wells with data are active (visible). The lower right (“Report window”) reports statistics for each channel and each well; Columns denote plate coordinates, channel, Cq values calculated using the (automatically set) threshold (“CqTh”) or a method based on second derivatives (“Cq2d”), and (raw) fluorescence signal min and max values.



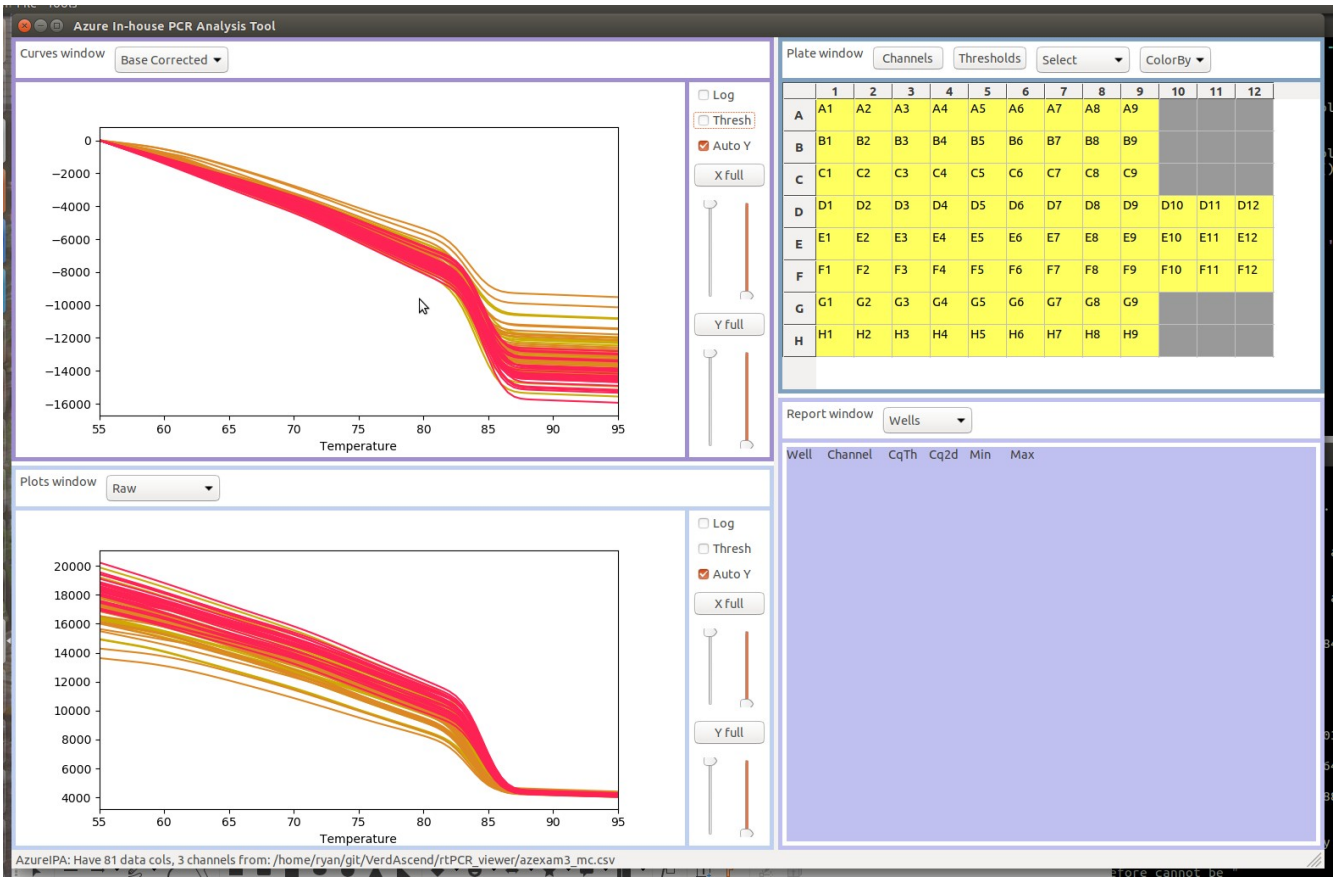
3) Select subset of wells. To do this, first choose “None” in the select pull-down in the “Plate window” to deactivate all cells. Then select a few cells by clicking and pulling on them; Cells may alternatively be chosen a row or column at a time by clicking on the row / column labels. The screenshot shows three active wells; these happen to be in different channels and so are plotted in different colors (the channels for each well are listed in the “Report window”).



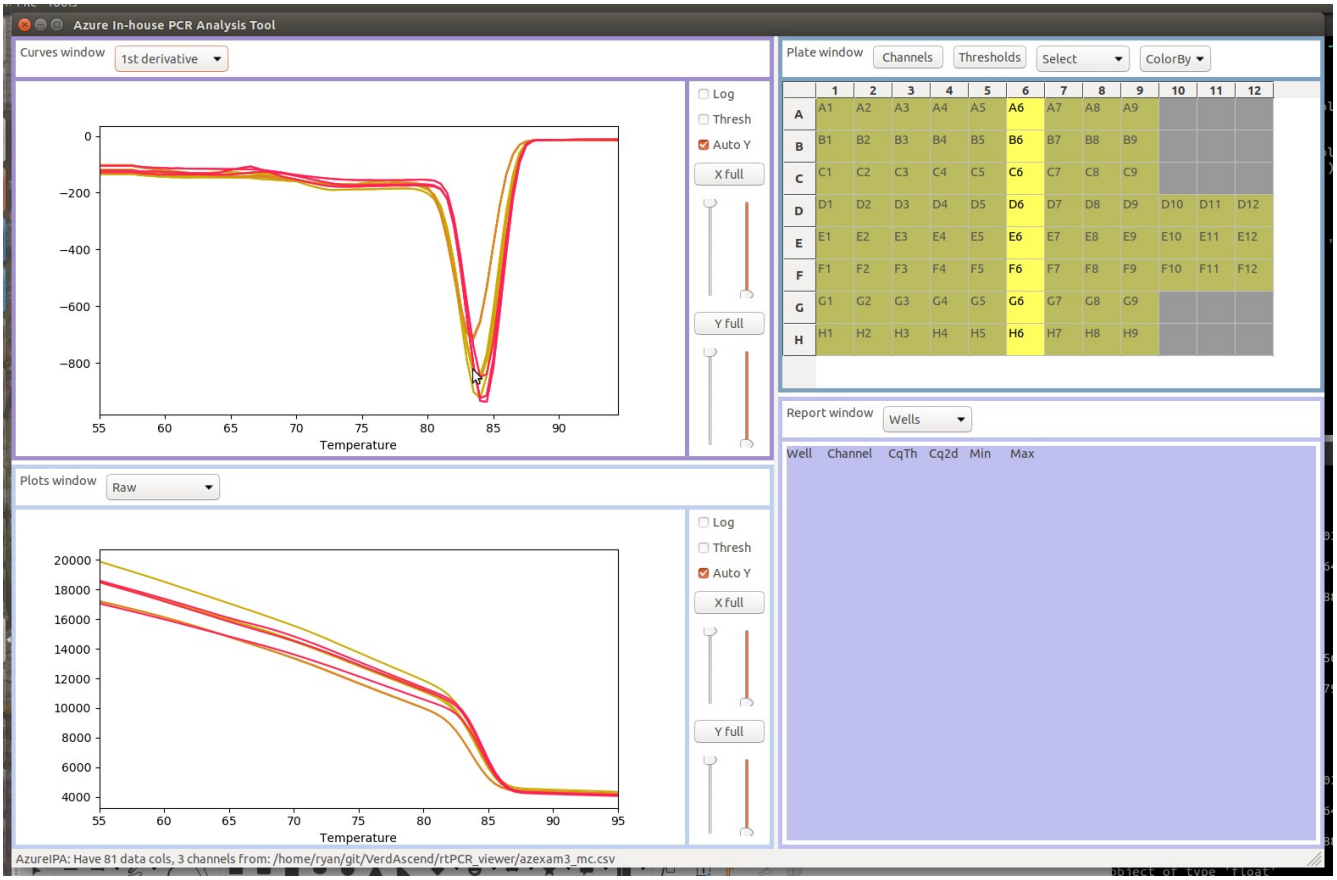
4) Change view to log scale and add thresholds. Select a column of wells (which includes three channels). Click on “Thresh” button to display the threshold as dotted lines. To adjust threshold values, click on the “Thresholds” button in the “Plate window” to bring up per-channel sliders. Note that changing thresholds changes Cq values (“Report window”) in addition to the visible positions in the plots.



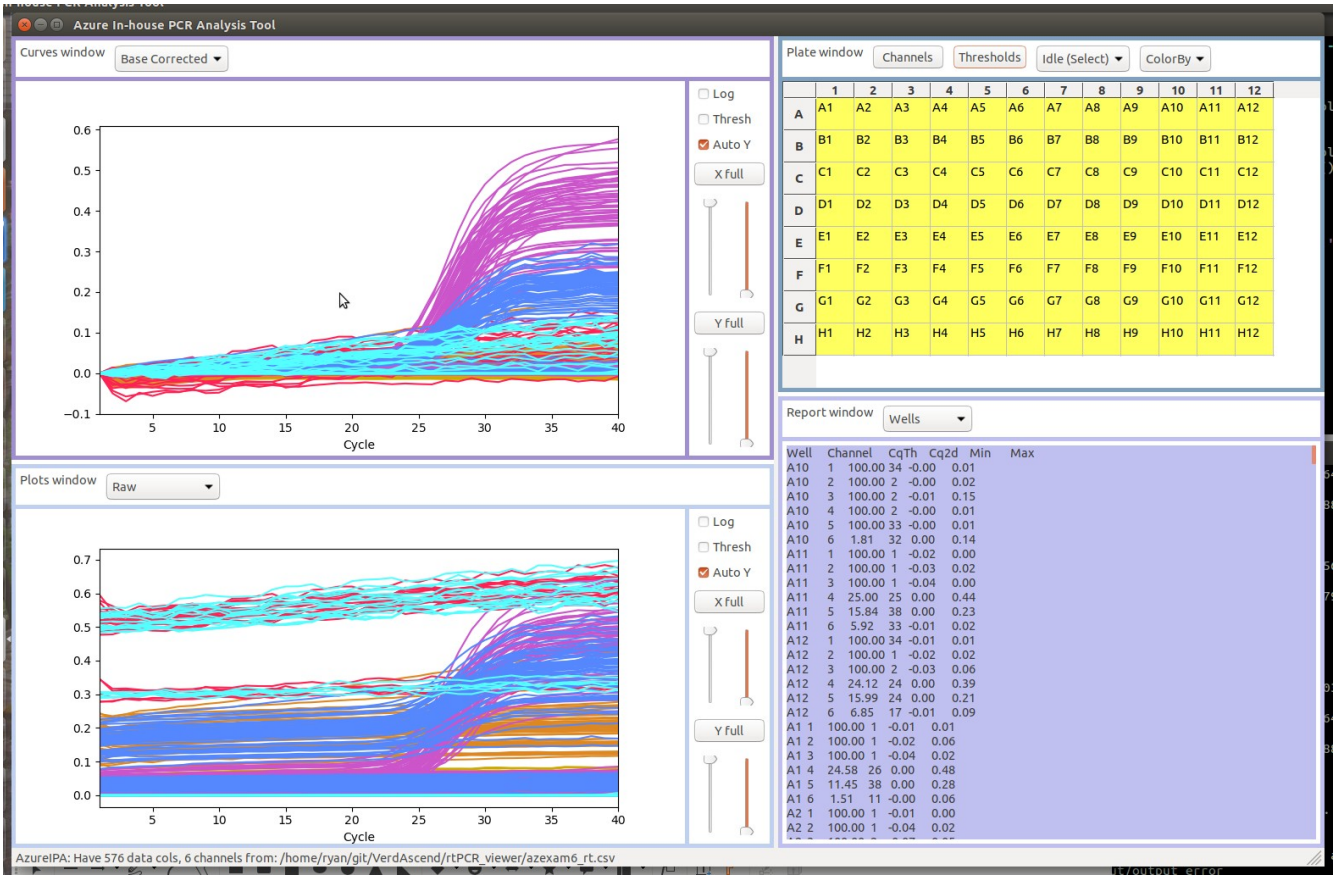
5) Load melting data. Open > Data then specify “`azexam3_mc.csv`”. Default view is all wells.



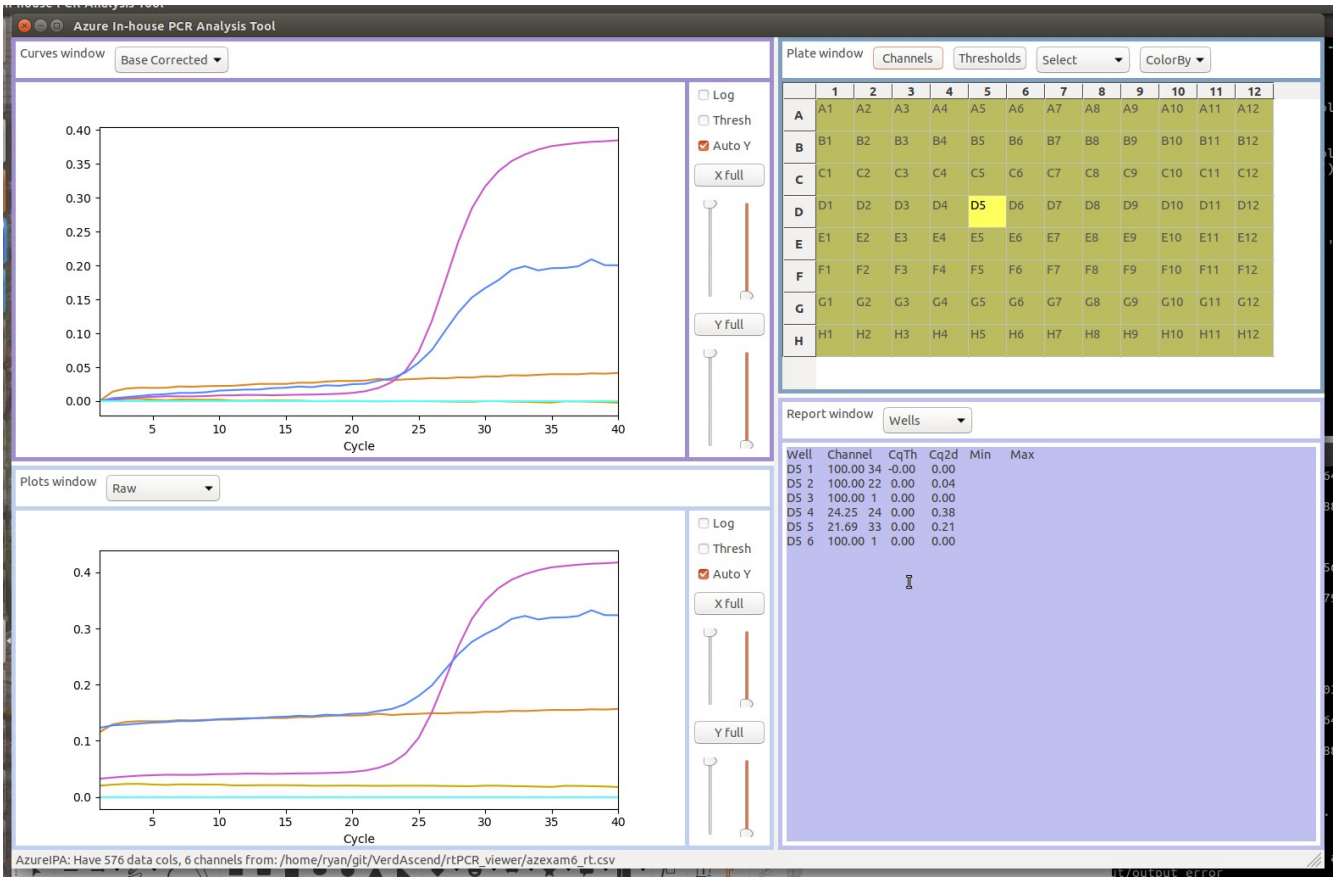
6) Look at melting data first derivatives for a subset of wells. Select a subset of wells via column. In the (top) plot pull-down, choose “1st derivative”. For melting data, the maximum dip in first derivatives is considered the “melting temperature” and may be used to characterize a PCR product.



7) Load dataset with six channels per well. Open > Data then specify “`azexam6_rt.csv`”. Default view is all wells, all channels (loading may be somewhat slow).



8) Look at single well with multiple channels. Use the “Channels” pop-up to select which channels are active.



9) Change preferences. Under “Tools” select “preferences” to get a pop-up where various attributes can be changed. Adjustable attributes are primarily colors. Select the attribute via the pull-down menu, then select “Edit” to change. Use “Apply” or “Cancel” to keep or discard changes.