Microarray QC Workflow

For questions, please contact Mark Fisher (fishema@ohsu.edu or mark.aaron.fisher@gmail.com).

Even if I don't work here anymore, please email me with troubleshooting questions. I'd rather have this tool be useful than a broken, discarded thing.

Setting the Workflow Up

Download

If you are a member of Bcore, clone the IGLTools repo. from our Gitlab instance: `git clone http://gitlab.bcore.ohsu.edu/bcore/IGLtools.git`.

There is also a working version on Box at https://ohsu.box.com/s/7xgreyyd4lji6xh5yubeucgx6elfxil9 (last updated 12/1/2016).

And a video tutorial in that same Box folder at https://ohsu.box.com/s/78gi3mjwf38gjboasty3ni4b43loadrk (last updated 12/1/2016).

Install Python and R

Mac

Python should come pre-installed. Test this by opening Terminal and typing, "python" or "python2"; if there are no errors, python is already installed.

R should be installed from its download page. Test that the installation worked as expected by opening Terminal and typing, "R"; if there are no errors, R was correctly installed.

Windows

R and python did not come pre-installed on the Windows machines in the Integrated Genomics Laboratory. You can install python by going to the python downloads page.

You can install R from its download page.

First-time configuration

Mac

You may need to change the paths for Rscript and python to those on your local machine. It's quite possible that those will vary from machine to machine.

In order to do so, navigate to and open Master_run.sh in the IGLtools directory in some kind of text editor.

Add the whole path of the Rscript and python programs to the .bat file. **Note that the python and Rscript may be called multiple times in this .bat file**. By default, these paths are:

"/usr/local/bin/Rscript"

and

"/usr/local/bin/python2"

Note that the quotes may be necessary if your path contains any white space.

Windows

You may need to change the paths for Rscript.exe and python.exe to those on your local machine. It's quite possible that those will vary from machine to machine.

In order to do so, navigate to and open Master_run.bat in the IGLtools directory in some kind of text editor.

Add the whole path of the Rscript.exe and python.exe programs to the .bat file. **Note that the python and Rscript may be called multiple times in this .bat file**. By default, these paths are:

"C:\Program Files\R\R-3.3.2\bin\Rscript.exe"

and

"C:\Python27python.exe"

Note that the quotes may be necessary if your path contains any white space.

Using the GUI

Input from other programs

1. This guide assumes that you have already run Expression Console and saved the full non-normalized report and in some cases the normalized report in accordance with the Affymetrix expression array SOPs in iglshare.

Open the GUI

1. Navigate to MicroarrayQC_GUI/dist/MicroarrayQC_GUI.jar and double-click to open it. You'll notice that there are three main panels: Input variables, Execute, and Transfer to Investigator Ready.

Input Variables

1. Select a project directory by clicking on the Browse button corresponding to the project directory. Click open when you have selected the relevent folder. *The project directory should be the folder that contains the ARR, AUDIT, CEL, etc. files.* This will very likely be the project folder of interest in \wingmsr\Data_prep.

2. Select a destination directory by clicking on the Browse button corresponding to the destination directory. Click open when you have selected the relevent folder. In almost all cases, *the destination directory should be the same as the project directory*.

3. Select an Expression Console non-normalized report text file. Click open when you have selected the relevent file. This file should in almost all cases live inside the project directory designated above.

4. Select an Expression Console normalized report text file. Click open when you have selected the relevent file. This file should in almost all cases live inside the project directory designated above. Some array types don't warrant this, in which case you can simply click Cancel after clicking the Browse button. Note: you have to at least click the Browse button and Cancel. Not doing anything at all with this input will result in an error.

5. Type in the project ID. Include the "*Set#" if warranted. In most cases, the project ID will be the same as the name of the project folder (i.e., \wingmsr\Data*prep\ProjectID).

6. Point the GUI in the direction of the directory containing all of the scripts that the GUI will use. These scripts will typically live in the same directory containing the MicroarrayQC_GUI folder (i.e., they should all be packaged together).

Execute

1. If you're running the .jar file from a Mac, click, "Run (mac)". If you're running the .jar file from a Windows machine, click "Run (windows)".

2. The output will be generated in the designated destination directory. Expect 7 output files:

    - **PathToTheExpressionConsoleNonNormalizedReport_processed.csv**: the Array Performance Summary precursor. Should contain all of the information needed excepting meta-data and color-

coding.

- **ARR_info.txt**: a tab-delimited table containing each sample's file name, whole barcode, lot number, and hyb. date. Please note that this "hyb date" may not be the actual date of hybridization, as the ARR file generation does not always occur on the same day a hybridization. Ask the technician in charge of hyb. to clarify that these are correct and edit in the APS if not.
- **AUDIT_info.txt**: a tab-delimited table containing each sample's file name, the fluidics station ID that did the washing for that sample, the fluidics module ID, the fluidics protocol, and the scanner serial number.
- **AUDIT_info_rescan_catcher.txt**: contains the same information above, but captures every rescan (meaning that the number of columns in each row will not necessarily be the same for each sample). Where AUDIT*info.txt only reports the details of the first scan for each sample,* *AUDIT*info*rescan*catcher.txt reports all of them.
- **masterErrors.txt**: a report that compiles all of the text directed to standard error during the runs of all of the subscripts. If something went wrong, you would find out about it in this file.
- **masterOutput.txt**: a report that captures all of the text directed to standard output during the runs of all of the subscripts. This output will contain useful information pertaining to duplicated sample names (or duplicated subcomponents of the names). GPSR names and User IDs, for instance, should have the same number of items and unique items. If they don't, there are duplicated names. If the audit file and the rescan catcher files are different, this will be reported here as well.
- **samplist.csv**: a csv-formatted table containing all of the sample names. This list can be edited and potentially passed to another tool (not incorporated as of 11/16/2016) that automatically generates the Investigator_ready project directory and populates it with array-related files for only the samples in this list.

Transfer to Investigator Ready

1. This panel will allow you generate a project folder in the Investigator_ready directory, replete with all of the standard subfolders for a typical Affymetrix array project coming through the GPSR to be delivered to a client. The folder's AGCC subfolder will be populated with all array-associated files (.CEL, .ARR, .AUDIT, .JPEG, etc.) for *only those samples that are listed in samplist.csv in the destination directory specified in the Input Variables panel earlier*. This means that if the GPSR decides not to provide the client with a certain sample due to poor performance, this sample should be removed from samplist.csv *before* running this panel.

2. Select the Investigator_ready directory by clicking, "Browse". Click open when you have selected the relevent directory.

3. If you're running the .jar file from a Mac, click, "Run (mac)". If you're running the .jar file from a Windows machine, click "Run (windows)".

4. The output will be generated in the designated destination directory. The folder structure will look like this:

```
ProjectID

├── Data_Prep_QC

└── Genome

    ├── AGCC (array-associated files will be in here)

    └── Absolute_Data
```

Under the hood: what the workflow comprises

MicroarrayQC_GUI: A folder containing a NetBeans (v.8.2) project used to generate the microarray QC GUI. The GUI wraps the tools below into one user-friendly tool that allows users to input arguments required by the subscripts graphically.

In the \dist\ subdirectory, you'll find a .jar file. In the Execute panel, the wrapping occurs by running either a bash shell script for Macs (Master_run.sh) or a batch script for Windows machines (Master_run.bat) that takes all of the arguments supplied by the user and feeds them as arguments to the various subscripts below:

Duplication Catcher: Python script that errors out and reports if the GPSR names contain duplicates. File name is gpsr_dup_catcher.py.

AUDIT and ARR Parser: Python script that pulls out hyb date, fluidics ID, fluidics protocol, etc., etc. from ARR and AUDIT files, reports as csv-formatted spreadsheets. File name is array_data_extractor.py.

APS Report Generator: R Script that takes the tabular QC data from (an) Expression Console run(s), subsets certain columns depending on array type, and combines with ARR and AUDIT data to generate a csv-formatted spreadsheet that is the precursor to current APSs. File name is 1_APS_generator_master.R.

In the Transfer to Investigator_Ready panel, the wrapping occurs by running either a bash shell script for Macs (transferToInvestigatorReady.sh) or a batch script for Windows machines (transferToInvestigatorReady.bat) that takes all of the arguments supplied by the user and feeds them to:

Investigator_ready Project Directory Generator and Transfer Script: Python script that generates a standard, client-ready project folder, including subdirectories, and then populates the AGCC subfolder with the array files corresponding to samplist.csv in the destination directory from the Execute panel. File name is copy*from*this_list.py.