Jeff Mecham

IS 520 Final Project

Executive Summary

The BYU Department of Nutrition enlists students as research assistants from a variety of majors. Research assistants are generally looking for lab experience necessary for graduate programs. This project was created as a solution to a question in Dr. Jason Kenealey's lab. Dr. Kenealey's lab focuses on understanding the biochemical mechanisms and effects of naturally occurring molecules on different cancers. Dr. Kenealey's research assistants use many different machines to assist their research that eventually export an Excel file with results after the experiment is completed.

After finishing an experiment, students will take the readout exported by the machine, format a new workbook, copy and paste data from the readout to the new workbook, write in formulas to analyze the data, and finally create graphics that clearly convey the data. The process is cumbersome and repetitive, all while leaving plenty of room for errors that skew data.

I recently started running experiments using the Polymerase Chain-Reaction (PCR) ThermoCycler. The objective of the experiment is to determine the quantity of transcriptions or copies of genes that are being expressed. Cells treated with different pharmacological agents are lysed open to examine mRNA content, which is an indication of genes being expressed. After a few steps the mRNA is converted to more stable cDNA and quantitative PCR (qPCR) is run against each of the different treated-cellular samples. qPCR uses different primers to target specific genes of interest and amplify the number of gene transcripts to readable levels. In order to control for differing levels of overall cDNA content between different treatments, scientists use an internal standard gene as a reference. The gene expression of this internal standard is consistent regardless of treatment options. The change in expression of the target genes is important to understand because expression of different genes indicates a change in biological function. For example, cancer cells often decrease the amount of important tumor-suppressing proteins (like p53) and the targets of those tumor-suppressing proteins (like TP53INP, PUMA, NOXA, etc.). The ability of pharmacological agents to upregulate (or in some cases downregulate) the target genes of interest can be an important mechanism of their efficacy in treating disease.

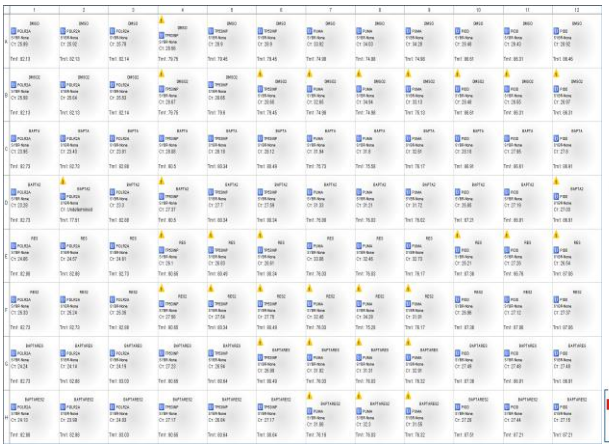
My system is designed to first prompt the user to choose the file path of the readouts to be analyzed (with the option to select an internal standard from a previous experiment on a separate workbook). Then, the procedure accepts input from the user regarding the control samples, the internal target gene, and the target gene of interest. After, it analyzes the different readouts from the PCR machine to determine the values of ΔCt(control), ΔCt(treated), fold change, and standard deviations for each sample, important values in determining the number of transcripts in each gene. Ultimately, this data is converted into an easy to read bar chart. The system is designed to expand or contract to fit different amounts of sample, target, and replicate numbers, and should accommodate any other students running qPCR from the same machine. The output of my system is a workbook with a copy of the original readout from the ThermoCycler along with a new summary sheet that contains a formatted table summarizing the above stated values and chart.

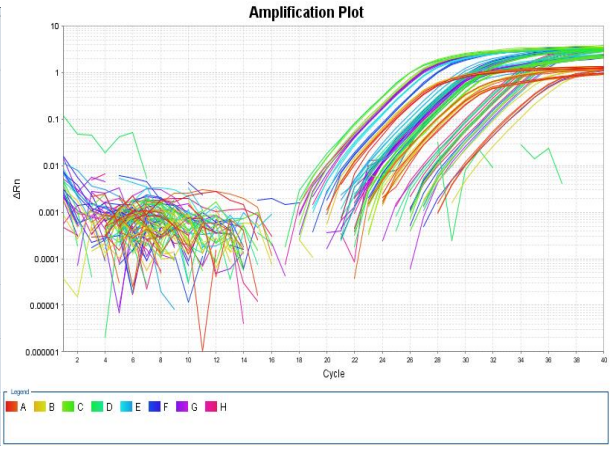
Project Details and Procedure Description

Ct Values and Readouts from ThermoCycler

qPCR is run in a 96-well plate, with each well experiencing its own reaction conditions. Generally, samples and targets are run in triplicate to account for variability. In the figure below on the left is an example of the plate setup on the qPCR software. Research assistants will tag each well with the sample being tested and the primer being used to target a specific gene. This information is crucial because it is put directly into the readout provided at the conclusion of the experiment.

The figure below and to the right shows an example of the amplification plot of a qPCR reaction. Outside of the beautiful rainbow it creates, this graph is used to determine Ct values, which are the main measurement of the qPCR. Because the reaction is exponential in its growth, each sample experiences a steep increase at a designated cycle as indicated by the x axis. The time that the steep increase occurs is referred to as the Ct value, and is an indirect measurement of the number of transcripts found in particular cellular sample.



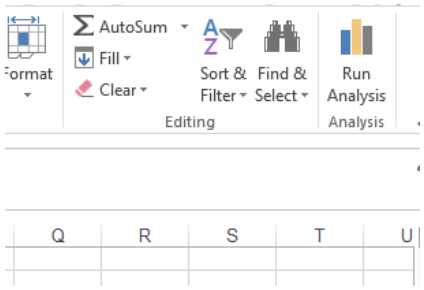


After the experiment is completed, the results can be output into a convoluted excel readout. Below is a section of one of the readouts to give an idea of the data that will be analyzed for this project. Although values like Ct mean and Ct standard deviation are calculated, they are not normalized to an internal reference, or to a non-treated control, and therefore do not provide the information necessary to really interpret the data.

	A	B	C	D	E	F	G	H	I
1	Block Type	96well							
2	Chemistry	SYBR_GREEN							
3	Experiment File Name	D:\Applied Biosystems\StepOne Software v2.3\experiments\Jeff Mecham BAPTA Res Run 4.eds							
4	Experiment Run End Time	2016-03-28 18:52:10 PM MDT							
5	Instrument Type	steponeplus							
6	Passive Reference	ROX							
7									
8	Well	Sample Name	Target Name	Task	Reporter	Quencher	Ct	Ct Mean	Ct SD
9	A1	DMSO	POLR2A	UNKNOWN	SYBR	None	25.89363	25.86522	0.074732
10	A2	DMSO	POLR2A	UNKNOWN	SYBR	None	25.92158	25.86522	0.074732
11	A3	DMSO	POLR2A	UNKNOWN	SYBR	None	25.78045	25.86522	0.074732
18	A10	DMSO	PIDD	UNKNOWN	SYBR	None	28.47642	28.60805	0.268206
19	A11	DMSO	PIDD	UNKNOWN	SYBR	None	28.43109	28.60805	0.268206
20	A12	DMSO	PIDD	UNKNOWN	SYBR	None	28.91664	28.60805	0.268206
21	B1	DMSO2	POLR2A	UNKNOWN	SYBR	None	25.58162	25.58541	0.053707
22	B2	DMSO2	POLR2A	UNKNOWN	SYBR	None	25.64091	25.58541	0.053707
23	B3	DMSO2	POLR2A	UNKNOWN	SYBR	None	25.53369	25.58541	0.053707
24	B4	DMSO2	TP53INP	UNKNOWN	SYBR	None	28.66598	28.65579	0.009166
25	B5	DMSO2	TP53INP	UNKNOWN	SYBR	None	28.64822	28.65579	0.009166
26	B6	DMSO2	TP53INP	UNKNOWN	SYBR	None	28.65316	28.65579	0.009166
27	B7	DMSO2	PUMA	UNKNOWN	SYBR	None	32.64863	33.57285	1.208384
28	B8	DMSO2	PUMA	UNKNOWN	SYBR	None	34.94025	33.57285	1.208384
29	B9	DMSO2	PUMA	UNKNOWN	SYBR	None	33.12966	33.57285	1.208384
30	B10	DMSO2	PIDD	UNKNOWN	SYBR	None	28.4797	28.66595	0.262195
31	B11	DMSO2	PIDD	UNKNOWN	SYBR	None	28.55236	28.66595	0.262195
32	B12	DMSO2	PIDD	UNKNOWN	SYBR	None	28.96578	28.66595	0.262195

Connect Entire Procedure to Ribbon Button

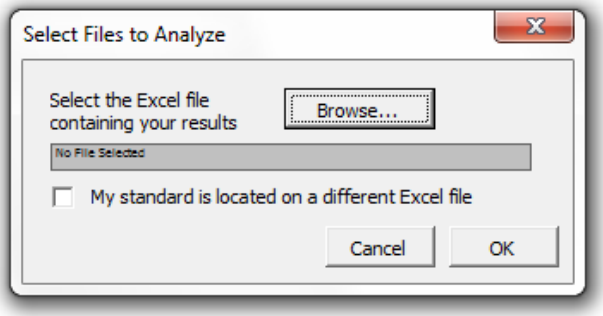
In an effort to make the navigation of the procedure as painless as possible, I attached a "Run Analysis" button onto the home tab of the Excel ribbon in my workbook.

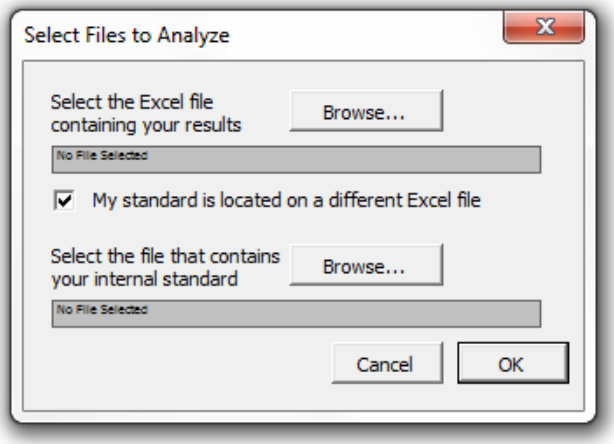


Form1: Select Readout File(s) to Analyze

Overview of Form

When the user click the RibbonButton, frmFileSelect is initialized. This form is designed with the command button "Browse…" that opens the msoFileDialogFilePicker when clicked showing only Excel spreadsheet options. If the user pick a file, the file is input into the box below that currently says "No File Selected."





If the user wishes to use an internal standard from a previous experiment stored in a different workbook, the user can check the box stating that the standard is located in a different Excel file. This extends the form and gives a similar option as that above to select the file that contains the internal standard.

These workbooks are assigned to variables that are used throughout the rest of the code to distinguish between the two. If no second file is selected, the second variable is set equal to the first variable, and one workbook is used for the entire procedure.

Form 2: Analysis Setup

Create Setup Sheet to Populate Analysis Form

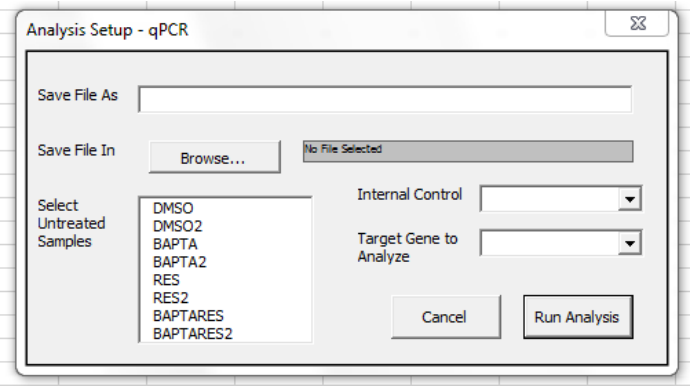
Because each experiment has a different design with different objectives, this procedure requires a few important inputs from the user. Before initializing the second form, a new sheet is created in the

workbook containing the results from the most recent experiment and is named “Setup.” This sheet is designed to run a loop through both workbooks’ results sheets in order to determine the names of different samples used and the different gene options to be used for the target gene and internal standard. In order to ensure each option appears in a column only once to correctly populate the form, I used the application function countif. The end result of this procedure returns three columns: sample names (from current experiment), internal standard gene options (from either the current experiment or a previous experiment if selected in the file select form), and target gene options (from current experiment).

Samples	Internal Standard Gene Options	Target Gene Options
DMSO	NOXA	POLR2A
DMSO2	PIDD	TP53INP
BAPTA	POLR2A	PUMA
BAPTA2	TP53INP	PIDD
RES		
RES2		
BAPTARES		
BAPTARES2		

Overview of Form

This form is crucial for the analysis in the next step.



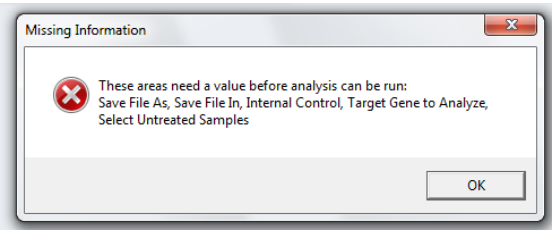
Select Untreated Samples. On the left, a listbox with multiselect capabilities prompts the user to select all untreated (or control) samples. This is important because those control samples will be used to normalize data from all of the other samples.

Internal Control and Target Gene to Analyze. These combo boxes populate from the setup sheet and allow the user select the internal control gene and target gene from the drop down menu.

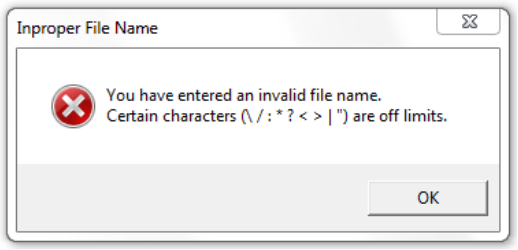
Save File As and Save File To. The end results of this experiment will save a new workbook to the specified location with the specified file name.

Data Validation

Each input in this form is essential for the analysis. If the user leaves any of the fields empty or untouched, and pushes run, a message box informs the user to fill out the proper information before continuing forward.



Inputs in the Save File As section are subject to validation of proper characters for file names. If the user uses any of the characters listed in the figure to the right, the file name would not be valid. This message box informs the user they must use proper characters for the file name.



Analysis and Calculations

Overview

The real bulk of this procedure comes during the analysis phase. After data is imported properly and the user click "Run Analysis," the procedure begins to analyze the data. The first sub procedure formats the table that the data will be input into. The table design is dynamic and will adjust to fit the number of samples depending on the data being analyzed. The sub procedure "figureFoldchange" does the hard number crunching. After all the needed values are calculated, they are placed into the data table on the sheet named "summary" in the original readout workbook to eventually be saved as a new file for interpretation.

Formulas

The value of interest is known as fold change. Fold change is a normalized value that expresses if a gene is overexpressed (>1) or under-expressed (<1). In the past, these calculations involved a lot of head scratching and excel work to figure out what data to include.

The important formulas for this analysis are:

ΔCt (control) = Ct(Target Gene: control) – Ct (Internal Gene: control)
ΔCt (treated) = Ct(Target Gene: treated) – Ct (Internal Gene: treated)
ΔΔCt = ΔCt (treated) - ΔCt (control)
Fold Change = 2^(-ΔΔCt)

Calculations and Methodology

ΔCt (control). This value is the same for each treatment, and is ultimately stored on the summary sheet in "C4." In order to calculate this, the data from the listbox of the analysis form is first imported into an array of string variables and eventually converted into one string separated by commas. This string argument contains the names of the samples that were selected as controls from the analysis form, and will be used in determining what Ct values to average. The procedure then runs through a loop with a running sum and count that is added to whenever the sample of the line matches at least one of the controls selected in the form *and* the target gene of the line matches the internal standard provided in the form. The screen grab below shows the example of values placed into the last column when the sample "DMSO" is selected as a control and "PIDD" is selected as the internal standard. Note that because the target of the other lines differs from "PIDD," no calculation takes place in the last column. The average of all of the control samples for the internal is set to a variable InternalControlAverageCT. The same procedure is completed for the target gene and then ΔCt (control) is calculated as the second average minus internalControlAverageCT.

	Sample Name	Target Name	Cт	Cт Mean	Cт SD	ΔCT Control
8						
9	DMSO	POLR2A	25.89363	25.86522	0.074732	
10	DMSO	POLR2A	25.92158	25.86522	0.074732	
11	DMSO	POLR2A	25.78045	25.86522	0.074732	
18	DMSO	PIDD	28.47642	28.60805	0.268206	2.42127864
19	DMSO	PIDD	28.43109	28.60805	0.268206	2.37594859
20	DMSO	PIDD	28.91664	28.60805	0.268206	2.86149851
21	DMSO2	POLR2A	25.58162	25.58541	0.053707	

ΔCt (treated). This value changes for each of the different treatment options and is stored into column "C" on the summary page. First a similar loop is run on the sheet containing the internal target sheet, but instead of averaging the Ct values of control treatments, the loop moves down the treatment options one at a time. I used a nested for loop to run the overall calculation for each sample on the summary page. The average ct for the internal gene of interest for each treatment was calculated and then subtracted from the gene of interest. The main variation from the previous loops was that this loop had to run for each sample and took in different inputs, but the methodology was similar.

ΔΔCt, Fold Change, and Standard Deviation. ΔΔCt is calculated with ΔCt (treated) - ΔCt (control). However, in order to calculate the standard deviation of the fold change, ΔΔCt had to be first calculated for each individual line of the results sheet, and then averaged into the summary sheet. Fold change was calculated in a similar fashion on each line by using the formula stated above, and standard deviation was calculated manually with general arithmetic in a loop. This version of the code is fairly representative of many of the loops run in my procedure. In the results sheet, columns are appended onto the end of the data for ΔΔCt and Fold Change. Below is a section of code that gives a peek into its methodology.

```
'DETERMINE DELTA CT TREATED FOR EACH TREATMENT AND INPUT INTO SUMMARY SHEET
z = 7
Do Until summary.Cells(z, 2).Value = ""
    treatment = summary.Cells(z, 2).Value
''FIRST DETERMINE AVERAGE CT OF INTERNAL FOR TREATMENT
    r = 9
        Do Until results2.Cells(r, 2).Value = ""
            If UCase(results2.Cells(r, 3).Value) = UCase(frmAnalysis.cboInternal.Value) And UCase(results2.Cells(r, 2).Value) = UCase(treatment) _
                                                And IsNumeric(results2.Cells(r, 7).Value) Then
                sum = sum + results2.Cells(r, 7).Value
                count = count + 1
            End If
            r = r + 1
        Loop
    InternalTreatedAverageCT = sum / count
    sum = 0
    count = 0

''DETERMINE DELTA CT TREATED VALUES FOR THIS SAMPLE AND ASSIGN TO RESULTS SHEET AND PLACE INTO COLUMN Z
    r = 9
        Do Until results.Cells(r, 2).Value = ""
            If UCase(results.Cells(r, 2).Value) = UCase(treatment) And UCase(results.Cells(r, 3).Value) = UCase(frmAnalysis.cboTarget.Value) _
                                And IsNumeric(results.Cells(r, 7).Value) Then
                results.Cells(r, "Z").Value = results.Cells(r, 7).Value - InternalTreatedAverageCT

            ''COUNT AND SUM FOR DELTA CT TREATED AVERAGE FOR SUMMARY SHEET
                sum = results.Cells(r, "Z").Value + sum
                count = count + 1

            ''PUT IN DELTA DETLA CT VALUE IN RESULTS SHEET COLUMN AA
                results.Cells(r, "AA").Value = results.Cells(r, "Z").Value - summary.Cells(4, 3).Value
            ''COUNT AND SUM FOR DELTA DELTA CT TREATED AVERAGE FOR SUMMARY SHEET
                sum2 = results.Cells(r, "AA").Value + sum2

            ''PUT IN FOLD CHANGES INTO RESULTS SHEET AB
                results.Cells(r, "AB").Value = 2 ^ (-1 * results.Cells(r, "AA").Value)
                sum3 = results.Cells(r, "AB").Value + sum3
            End If
            r = r + 1
        Loop
```

```
        ''PUT DELTA CT TREATED INTO SUMMARY SHEET
        deltaCTtreated = sum / count
        sum = 0

        summary.Cells(z, 3).Value = deltaCTtreated

        ''PUT DELTA DELTA CT INTO SUMMARY SHEET
        DeltaDeltaCT = sum2 / count
        sum2 = 0
        summary.Cells(z, 4).Value = DeltaDeltaCT

        ''PUT FOLD CHANGE INTO SUMMARY SHEET
        foldchange = sum3 / count
        sum3 = 0
        count = 0
        summary.Cells(z, 5).Value = foldchange

        ''CALCULATE STANDARD DEVIATION AND PUT INTO SUMMARY SHEET
        r = 9
        Do Until results.Cells(r, 2).Value = ""
            If UCase(results.Cells(r, 2).Value) = UCase(treatment) And UCase(results.Cells(r, 3).Value) = UCase(frmAnalysis.cboTarget.Value) _
                    And IsNumeric(results.Cells(r, 7).Value) Then
                variance = variance + (results.Cells(r, "AB").Value - foldchange) ^ 2
                count = count + 1
            End If
            r = r + 1
        Loop

        stDev = (variance / (count - 1)) ^ (1 / 2)
        summary.Cells(z, "F").Value = stDev
        count = 0
        variance = 0

    z = z + 1
Loop
```

Create Chart and Save As

The final section of the procedure creates a chart using the sample names and fold change values on the summary sheet. The file name specified in the analysis form is used to save the results into a new workbook in the specified location. All of the workbook used for analysis are closed without any of the changes made during the procedure being saved so as not to corrupt further data.

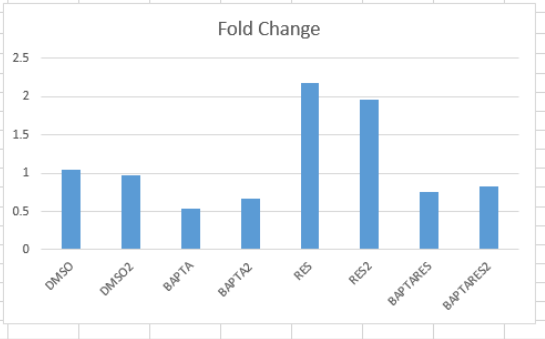
In the end, the user is left with two open workbooks, one bearing the original code to run the analysis and the other with a save version of the output of the machine. The results are easy to interpret, and the entire process takes less than a minute!

Below is an example of the finished summary sheet.

Target Gene	PIDD	Internal Control	POLR2A

Δ CT (Control)	2.581856092

Sample Name	Δ Ct (Treated)	ΔΔ Ct	Fold Change	St Dev
DMSO	2.534645716	-0.047210375	1.044789564	0.1832
DMSO2	2.629066467	0.047210375	0.978148925	0.1684
BAPTA	3.503744761	0.921888669	0.53094002	0.0699
BAPTA2	3.162002563	0.580146472	0.672005034	0.0793
RES	1.524499893	-1.057356199	2.174443442	0.747
RES2	1.624759674	-0.957096418	1.954679535	0.2762
BAPTARES	2.987360001	0.405503909	0.754977832	0.0035
BAPTARES2	2.863461812	0.281605721	0.825514514	0.0835



Assistance

Outside of a few points and tips from Dr. Allen, I received no additional help on this procedure.

Discussion of learning and conceptual difficulties encountered

This project was really eye opening. Originally, I thought the calculations would be simple. I found that the most difficult, yet rewarding, part of the project was actually figuring out the best way to approach writing the code. At first, I began by just typing everything that I thought should go in. By the end of my first draft, I had an experiment that calculated values, but it calculated the wrong values. I realized that in my haste to write the code, I went about the logic of the experiment entirely wrong. I remembered that in my IS 201 class our professor suggest writing a flow chart of the experiment design before jumping into the problem. After sketching the procedure onto a scratch sheet of paper, the work became a lot easier. I was able to methodically move through each step of the overall system.

I also feel that I learned a lot more about how actual vba code works. It is an obvious realization, but I learned to a much deeper extent the meaning of variables and how to work with them in different scopes. I feel much more comfortable in my ability to work with functions in order to accomplish all sort of different tasks.

One of the difficulties I ran into was working with the worksheet functions. I ended up bypassing a few of the worksheet functions I planned on using with the use of loops, if statements, and running sums/counts. Another limitation that I was able to overcome by actually hard coding the formulas in was standard deviation. I could not locate a “standard deviation if” formula, so I wrote the code myself.

Overall, this was a painstaking, but enjoyable experience. I feel much more confident at tackling other problems and challenges in the lab. I have a feeling that if I were to start this code from scratch, it would be much cleaner and a little less convoluted. I feel that that is part of the learning process.