WBS doc

qPCR Analysis
Summary:

RT-qPCR is a powerful technique that allows for scientists to measure gene expression changes quickly and accurately. It is often used as validations for RNA-seq data or can be used in combination with immunoprecipitations or polysome profiles to measure RNA-protein and translation of RNAs respectively. Although powerful and widely used, most analysis tools are in the form of excel sheets or on qPCR machines themselves. This increases the potential for user input errors and misinterpretations of data. To alleviate this, qPCR analysis will compute widely used types of analyses such as pfaffl and delta delta ct for users. Inputs will be Ct values and outputs will eventually be plots of your data with statistics.

1. **Module 1: Data processing and computation - completed**
    a. **Import_and_tidy_data**
        i. **Description:**
            1. This function will import your csv file and convert it to a useable dataframe for downstream analysis. It will do quality checks and preprocess your csv file by averaging triplicate Ct values and ensuring all of the headers are accurate.

        ii. **INPUT: File path to .csv file ; type of analysis**
        iii. **OUTPUT: data frame with Gene, condition, replicate, and average Ct values, will contain fraction or dilution if polysome or primer efficiency**

    b. **Primer_efficiency_calc**
        i. **Description:**
            1. This function will take the slope of your dilution series and convert it into a primer efficiency value – will be used in primer efficiency function, but also available to use if you have a slope already

        ii. **INPUT: slope of dilution series of Ct values**
        iii. **OUTPUT: primer efficiency percent**

    c. **Primer_efficiency**
        i. **Description:**
            1. A function that will take your dilution series dataframe and convert it to a primer efficiency for pfaffl analysis. Will check for proper dilution series (i.e 1/dilution) and proper Ct triplicates. Please use primer_efficiency parameter for import and tidy data. It will convert your dilution series to log and calculate the slope for that dilution series. Then it will use the primer efficiency calc function to return a primer efficiency.

        ii. **INPUT: dilution series dataframe**
        iii. **OUTPUT: modified dataframe, slope, and primer efficiency**

   d. **Delta_ct**
      i. **INPUT: Condition1, condition2, dataframe**
      ii. **OUTPUT: Ct condition 2 subtracted by Ct condition 1**
   e. **Delta_delta_Ct**
      i. **INPUT: Gene of interest, control gene, condition1, condition2, dataframe**
      ii. **OUTPUT: delta delta ct value**
   f. **Pfaffl**
      i. **INPUT: gene of interest, control gene, primer efficiency for GOI, primer efficiency control gene, condition1, condition2**
      ii. **OUTPUT: dataframe with average, stats, and replicate gene expression changes**
   g. **Polysome_analysis**
      i. **INPUT: dataframe from polysome profiles**
      ii. **OUTPUT: percentage of mRNA in each fraction as a dataframe with standard error of the mean**

2. **Module 2: Statistics – need to finish**
   a. **T-test**
   b. **ANOVA**
3. **Module 3: Plotting**
   a. **Primer_effiency_curve – completed**
   b. **Gene expression ratio (1 gene) – completed**
   c. **Gene expression ratio (many genes) – need to do**
   d. **Percent_in_polysomes – completed**

UPDATES:

1. Write functions for statistics module
   a. t-tests – can use help by scipy
   b. ANOVA – can use help by scipy

2. Update user input, make more modular and accept slight changes in input
   a. Use regex for headers
   b. Generate warnings when headers do not match
3. Give the user more ability to change plots
   a. Use if/else statements and get the user input when calling the plots
   b. Update documentation for each function.
4. Finish tutorial
   a. Show all three types of analyses
   b. Make it clear with and concise