Processing KO data

Step 1. Data normalization

	*Input files (required)***:**	*Example:*
1	map of the KO combinations	files/geneIndex.csv
2	map of gene indices to original gene name	files/geneNames.csv
3	list of lower matrix NaN gene indices	files/nan_lower_gene_index.csv
4	list of upper matrix NaN gene indices	files/nan_upper_gene_index.csv
5	raw data file for each condition	data/ 15C.csvand data/ No Treatment.csv

Script: 1_normalize_data.R
Libraries required: amap, gplots, extrafont, optparse

Usage:
Rscript 1_normalize_data.R -d [data directory path] -f [files directory path] -o [output directory path] -r [number of replicates] -w [number of wildtype matrix]

Example:
Rscript 1_normalize_data.R -d /Users/violahalder/Desktop/GitHub_GI_Analysis/data -f /Users/violahalder/Desktop/GitHub_GI_Analysis/files -o /Users/violahalder/Desktop/GitHub_GI_Analysis/output -r 3 -w 15

	Expected output files:	*Example:*
1	normalized data for each condition and replicate	output/allData.csv
2	average normalized data	output/avgData.csv
3	heatmap of average normalized data	output/ 15C_ODfull.pdf & output/ No Treatment_ODfull.pdf

Step 2. EPS (Genetic Interaction score) calculation

Script (jupyer notebook): 2_calculate_genetic_interaction.ipynb *Packages required*: numpy, pandas

	User-defined variables:	*Definition:*
1	cdn	which condition to process (15C or No Treatment)
2	num_reps	number of replilcates
3	main_dir	main directory path

Expected output file:
An Excel file per condition, with R+1 number of sheets, where R=number of replicates; last sheet is for average data.
Every sheet has the following columns: Gene_X, Gene_Y, Fitness_X, Fitness_Y, Fitness_XY, Fitness_YX, EPS_XY, EPS_YX
Average sheet has two additioinal columns - Average_Fitness_XY_Fitness_YX & Average_EPS_XY_EPS_YX

Example: 15C_fitness_and_eps.xlsx & No Treatment_fitness_and_eps.xlsx

Step 3. Run t-test to get p-values and check significant data for heatmaps

Script (jupyer notebook): 3_run_ttest.ipynb *Packages required*: numpy, pandas, scipy

	User-defined variables:	*Definition:*
1	cdn	which condition to process (15ºC or No Treatment)
2	num_reps	number of replilcates
3	main_dir	main directory path
4	Pval_cutoff	Change between 0.05, 0.01, or 0.001 depending of level of significance wanted
5	Eps_cutoff_pos	positive interaction eps cutoff; update the value as needed
6	Eps_cutoff_neg	Negative interaction eps cutoff; update the value as needed

Expected output files:

i. Excel files containing p-value data for the following comparisons:

   a) experimental vs actual_XY
   b) experimental vs actual_YX
   c) actual_XY vs actual_YX
   *experimental = Fitness_X * Fitness_Y*
   actual_XY = Fitness_XY
   actual_YX = Fitness_YX

   Examples:
   output/cdn1_pvalue_exp_vs_actual_XY.csv
   output/cdn1_pvalue_exp_vs_actual_YX.csv
   output/cdn1_pvalue_actual_XY_vs_YX.csv

ii. Excel file containing compiled data of above files

   Example: output/15C_pvalue_compiled.csv & No_Treatment_pvaluue_compiled.csv

Step 4. Generate heatmaps

Script (jupyer notebook): 4_generate_fitness_GI_significant_heatmap.ipynb

Packages required: numpy, pandas, scipy, seaborn

	User-defined variables:	*Definition:*
1	cdn	which condition to process (15ºC or No Treatment); can process both to generate hierarchical clustering
3	main_dir	main directory path

	Expected output files:	*Example:*
1	Fitness heatmap per condition	15C_Fitness_Score_heatmap.png & No Treatment_Fitness_Score_heatmap.png
2	Genetic interaction heatmap per condition	15C_Genetic_Interaction_heatmap.png & No Treatment_Genetic_Interaction_heatmap.png
3	Significant data heatmap per condition	15C_Significant_data_heatmap.png & No Treatment_Significant_data_heatmap.png
4	All_condition fitness hierarchical clustering	All_cdn_Fitness_Score_clustermap.png
5	All_condition genetic interaction hierarchical clustering	All_cnd_Genetic_Interaction_clustermap.png
6	Single deletion Fitness score heatmap	Single_Deletion_Fitness_Score_heatmap.png