

Last updated on: 25 September 2024

Prerequisites: Please ensure that Python version 3.10.6 (or above) is installed in your system. The following libraries must be installed in the same directory as python:

- pyreadr (version 1.16.0 or above)
- matplotlib (version 1.16.0 or above)
- numpy (version 2.0.0 or above)
- pandas (version 1.16.0 or above)
- scipy (version 2.0.0 or above)
- sklearn (*scikit-learn* version 3.1.0 or above)

execute the code '**mkfolders.py**' to ensure all folders are created before start collecting the data.

Note: After execution of *mkfolders.py*, several folders should be created in the parent directory of the current folder (as sibling directories of the 'main' folder). The data processed in subsequent steps will be saved in their respective locations, while all the plots generated will be saved in the 'Plots' folder.

Execution order:

For each of the following analysis, the codes must be run in the listed order. All data will be generated from '*fitness_data_wt.rds*' file (empirical data provided in Papkou et al's work).

A. Probability and number of fitness peaks in each size landscapes:

1. peak_count_functional.py
2. peak_count.py
3. peak_probability_functional.py
4. peak_probability.py
5. peak_plots.py
6. Joint_peak_plots.py

B. Fitness effect of individual mutations

1. mutation_effect.py
2. mutation_effect_plots.py

C. Distribution of fitness effects

1. DFE.py
2. DFE_3D_data.py
3. DFE_ks_test.py
4. DFE_mwu_test.py
5. DFE_stat_test_plots.py
6. DFE_3d_plot_functional.m
7. DFE_3d_plot_non_functional.m
8. DFE_2d_plot.py
9. DFE_expectation_plot.py

D. How neutral mutations change evolvability of a variant

1. mutation_evolvability_data.py
2. evolvability_plots.py

E. Epistasis in DHFR landscape

1. Epistasis_change_detailed_data.py
2. Epistasis_change_data_table.py
3. Epistasis_change_locus_plots.py
4. Epistasis_change_synonymous_mutations.py
5. synonymous_mutations_plots.py

F. Epistasis change in beta lactamase landscape

1. beta_lactamase_epistasis_change.py
2. beta_lactamase_epistasis_change_plots.py

G. Accessibility to highest peak through darwinian evolution

1. peak_accessibility_data.py
2. path_plots.py

Code details:

peak_count_functional.py : Finds and saves the data of number of high fitness peaks (fitness > 0.50774) in all n-size landscapes.

peak_count.py : Finds and saves the data of total number of (all) peaks in all n-size landscapes

peak_probability_functional.py : Finds and saves data of probability of finding a high fitness peak (fitness > 0.50774) in all size landscapes.

peak_probability.py : Finds and saves data of probability of finding (any) fitness peak in all size landscapes.

peak_plots.py : Generates and saves all plots for predicted (NK) and empirical fitness peak distribution in all landscape sizes.

Joint_peak_plots.py : Generates and saves plots containing combined empirical and NK predicted high fitness peak distribution.

mutation_effect.py : Tests all individual mutations on valid backgrounds (separately for functional / non functional / all backgrounds) and find saves their respective data. Also performs a linear regression on data of 'background fitness' and 'fitness effect of mutation' and saves the data of slope, intercept, the pivot point and the R² score.

mutation_effect_plots.py : Generates 'background fitness' vs 'fitness effect of mutation' plots for all mutations, as well as the bar plot of pivot point fitness of all mutations.

DFE.py : Finds the distributions of fitness effect for all variants, separates them in two categories - test data and prediction data for phenotypic DFEs and saves all data.

DFE_3D_data.py : Generates data for 3D phenotypic DFE plot and saves it as Matlab data file.

DFE_ks_test.py : Performs 2 parameter Kolmogorov–Smirnov test on upto 100,000 combinations of (test data) DFEs with similar background fitness to test reliability of phenotypic DFE.

DFE_mwu_test.py : Performs Mann-Whitney U test on upto 100,000 combinations of (test data) DFEs with similar background fitness to test the reliability of phenotypic DFE.

DFE_stat_test_plots.py : Generates and saves plots of p-value distributions of the KS and MWU tests.

DFE_3d_plot_functional.m ; DFE_3d_plot_non_functional.m : Running these codes would generate 3D plots of the phenotypic DFE in functional and non functional regime respectively (plots need to be saved manually)

DFE_2d_plot.py : Generates and saves 2D plot of superimposed (high fitness) phenotypic DFEs of different background fitness.

DFE_expectation_plot.py : Plots the expected (average) fitness effect of mutation in variants with similar background fitness.

mutation_evolvability_data.py : For each variant, finds a set of neutral mutations (with negligible fitness effect) and checks the relative increase / decrease in fitness effect of any mutation due to any specific neutral mutation.

evolvability_plots.py : Generates and saves box plot of effect on fitness effect due to a neutral mutation.

Epistasis_change_detailed_data.py : Generates and saves a detailed data of number of variants where a *<specific>* epistasis exhibited by a *<specific>* mutation pair changes to *<specific>* epistasis upon introduction of a *<specific>* mutation on the background. The resulting file would be too large to open directly on spreadsheet apps, and needs to be processed before being readable.

Epistasis_change_data_table.py : Compiles and saves tabulated data sets of - probability of epistasis change with given mutation pairs (rows) and the loci of the new background mutation involved (columns).

Epistasis_change_locus_plots.py : Generates and saves all plots for probability distributions of epistasis changing with mutation on each loci.

Epistasis_change_synonymous_mutations.py : Finds and saves data for instances where a synonymous mutation on an unaffiliated codon changes/does not change the nature of epistasis exhibited by a mutation pair on a variant.

synonymous_mutations_plots.py : Generates bar plots to find the percentage of cases for which a synonymous mutation on an unaffiliated codon

changes the nature of epistasis exhibited by a mutation pair on a variant.

beta_lactamase_epistasis_change.py : Collects and saves the data where a mutation on background changes the nature of epistasis due to a mutation pair in the beta lactamase landscape.

beta_lactamase_epistasis_change_plots.py : Plots the average number of instances for which the nature of epistasis in the beta lactamase landscape changes due to a mutation on the background.

peak_accessibility_data.py : Finds and saves data of the distribution of number/fraction of shortest paths which are accessible to the highest peak through darwinian evolution for variants at specific distances from the highest peak.

path_plots.py : Generates and saves stem plots for probability distribution of path accessibility of variants at a specific distance to the highest peak in shortest number of mutations.