Using the 'simplex_regression' script to analyze mutational epistasis

The 'simplex_regression.py' script will build a linear model of genetic interaction terms to assess the epistatic effects present in a protein. The model starts from the 1st order and automatically detect the highest, statistically significant epistatic order model and report its attributes. The following details the operation of the script.

Software requirements:

This example uses version 3 of Python and the following packages:

- Matplotlib
- Pandas
- NumPy
- SciKit-learn
- SciPy

To install these packages, simply use the pip-install command. For example:

```
pip-install matplotlib
```

Alternatively, you can install a version of Python 3 in the form of a package manager like Anaconda which will usually have all of these dependencies included in the base installation.

Data formatting:

This script only requires a single input file with the data generated from your experiments. The data can be edited in Excel or another spreadsheet program, and must be saved as a 'comma separated values' file (*a.k.a.* '.csv' file). The data file should contain 4 elements:

1. In the second row, provide the WT sequence in the first cell.
2. In the fourth row, provide a list of positions indicating which protein positions your dataset encompasses. In our example, we would use the following series: 233, 254, 271, 272, 306 and 313. Each position occupies a new cell in the row.
3. After the fifth row, provide a vertical list of genotypes in the first column, *i.e.* data that indicates the combination of amino acid states at each position. For instance, 'DHLLFI' for the first PTE variant on row 4. The header 'Genotype' needs to be present on row 5.

4. Note that after the fifth row, a vertical list of the observed functional measurements should be provided, *e.g.* 'fold change in butyrate esterase activity', in the second column corresponding to each genotype. The header 'Function' needs to be present on row 5.

An example of the input data is shown below in spreadsheet format. Please ensure that the format is *exactly* as shown below (Table 5), such that there are no gaps between the 'WT' (rows 1-2), 'Positions' (rows 3-4) and the 'Genotype' and 'Function' data (row 5 onwards).

Example of the input template

WT					
DHLLFI					
Positions					
233	254	271	272	306	313
Genotype	**Function**				
DHLLFI	-0.084				
DHLLFI	-0.013				
DHLLFI	0.059				
EHLLFI	0.298				
EHLLFI	0.387				
EHLLFI	0.417				
DRLLFI	1.648				
DRLLFI	1.529				

Running the script:

The script can be run from the command line using the following command:

```
python3 simplex_regression.py input_file.csv
```

Where 'python3 simplex_regrssion.py' initiates python to run the script and the name of the formatted input data file is provided immediately after. This generates three results files in the same directory as where the script was executed:

1. '*position_out.csv*' – Lists all effect coefficients of the highest order model for every combination of positions, and every order of interaction. It also contains the R^2 values for each coefficient.
2. '*genotype_out.csv*' – Lists all effect coefficients of the highest order model for every combination of positions, in every order of interaction, but with specific reference to the amino acid state (WT or mutant). Similar to '*position_out.csv*' with the key difference that each unique combination of amino acids is also considered as opposed to just the positions.
3. '*model_r2.csv*' – Lists the overall R^2 of the model at each order of interaction, as well as residual R^2 (ΔR^2).

In particular, the file '*position_out.csv*' includes a summary of all mutational effects as well as the R^2 provided by each parameter to the overall model. The results obtained in the output file '*position_out.csv*' can also be directly submitted into the 'visualize_epistasis.py' script to display the epistatic interactions on a crystal structure.