TWZLR : Daniel Carbonero, Chris Kuffner and Owen O’Connor

Background:

Traditional cloning can be labor intensive and time consuming especially when multiple parts need to be combined. However, performing homologous recombination in vivo can dramatically improve efficiency. Harnessing the short doubling time and natural competency of Vibrio Natriegens (V. Nat), the time and steps to clone may be reduced. V. nat allows for efficient homologous recombination of linear strands of DNA. By overlapping homologous arms, the parts may be added in a sequential order. This allows for the creation of a genetic circuit containing multiple parts from one pot. Layers may be added to create a multigene pathway that would allow for screening of a wide variety of combinations.

Software Components:

Setting configuration GUI (class init_config in TWZLR.py):

Component allows user to set parameters involving the design of the primers. Parameters are chromosome selection, circuit insertion range, length of homology, polymerase to be used, toggle switch for adapter use and selection of results output path. After the user selects save settings, the parameters will be written and passed to the main tool for sequencing. More details on the exact values each parameter can take are further detailed in the README.

Main Interface (class TWIZLR_int in TWZLR.py)

Component allows user to specify gene list according to layers and genes (indexes and subindexes). See video for further explanation on indexing system. Upon executing of functions in the main window, the list detailing the current gene list according to the specifications will be updated in the main window. The interface likewise handles data management of variables in the data console. This window also allows a user to re access and reconfigure settings. The ‘run tool’ function will run the sequencing tool for the displayed gene list.

TWIZLR.py

Main script containing all interfacing components and functions. Initializes interfacing, and, after user opts to ‘run tool’ will get specified DNA sequences for genes and chromosome homology arms at specified locations for primer design. The script will subsequently execute sequencing and graphing scripts to arrive at final results.

Semiglobal.py:

Contains Needleman Wunsch dynamic programming routine for semiglobal alignment of primers to longer sequences. Contains one function that takes two sequences (intended to be primer and longer part), a substitution matrix to arbitrarily score matching or mismatching alignments, and a gap penalty. Returns score for the final alignment (not the best one) an array F containing scores for all semiglobal alignments and a traceback matrix that enables users to find the sequences of the semiglobal alignments that produced a specific score at specific position F

Revcomplement.py

Contains revcomplemen function that generates the reverse complement of either a string or list of DNA base letters and returns it as a list. A second function complement is available but should only be used for display purposes when the user wants to portray double stranded DNA as two lines of text, one above the other with their bases complementary.

Local.py

Contains Smith Waterman dynamic programming routine for local alignment of one long DNA sequence to another (find the substrings in each sequence that best align). The smith waterman function takes the same inputs as semiglobal.py and returns the same outputs, but they are modified such that it is possible to trace back local alignments. Local.py also contains the function localtraceback that finds the alignment that produced a score at a specified position in the alignment score matrix. Finally, localtm predicts the highest possible melting temperature of the two different sequences annealing to one another.

graph.py

Makes a directed graph representing the combinatorial assembly of the various parts specified in the GUI.

sequencer.py

Designs primers to assemble three different parts using SOE PCR. The script uses semiglobal.py and local.py. There are many parameters that can be adjusted within, including the range of primer length, the range of sites where they can bing, and the substitution matrices and gap penalties used to score alignments. The script designs primers to amplify the three input sequences and add overlaps on their ends so they can be spliced together. Local alignment is used to set a threshold that primer and overlap melting temperatures must exceed to avoid SOE PCR failure by unexpected annealing of the three parts to eachother.

See README for detailed usage instructions