Functional Specification for crispr_array_generator

Willow Chernoske

Background

CRISPR arrays that encode for multiple CRISPR gRNAs allow for multiplexed gene editing. Cas12 systems are best suited for this multiplexed targeting, since they possess the power to process gRNAs from an RNA transcript with no additional inputs. Thus, the Escobar Lab and others harness these Cas12 CRISPR arrays for multiplexed targeting. However, designing these arrays can be tedious, as each gRNA within the array should be accompanied by separator, repeat, and annealing overhang sequences to optimize processing. With this, it is time-intensive and easy to make mistakes when designing these arrays by hand.

This tool can be used to check existing gRNAs or CRISPR arrays for common errors, and moreso, can be used to identify gRNAs from DNA and auto-process gRNAs into ready-to-order array oligonucleotides (oligos).

User Profile

The users of my tool will be people involved in research that want to build CRISPR Cas12 CRISPR arrays. This includes undergraduates, graduate students, post-doctorates, research technicians, and principal investigators. At a minimum, they will be able to access the DNA sequence they would like to target via CRISPR or be able to acquire gRNAs of interest, have a beginner-level knowledge of python, and be able to follow the guidelines provided.

Use cases

1. Check existing CRISPR gRNAs for common errors:
    a. Objective: The user wants to check chosen gRNAs for errors.
    b. User-system interaction: The user will use the python environment of their choice, install CRISPR_Array_Generator using pip, and then call the checkgRNA function. Here the user will input their gRNA(s) as python string(s) (one per gRNA) and run the function. The output will inform the user if any common errors are present such as non-DNA bases, CRISPR cut sites within the gRNA, or length errors (too long or too short).

2. Check existing CRISPR gRNA oligos for common errors:
    a. Objective: The user wants to check chosen gRNAs for errors.
    b. User-system interaction: The user will use the python environment of their choice, install CRISPR_Array_Generator using pip, and then call the checkArray function. Here the user will input all of their oligos (forward and reverse) as python strings (one per oligo) that will be assembled to create their array and run

the function. The output will inform the user if any common errors are present such as non-DNA bases, CRISPR cut sites within the array, gRNA length errors (too long or too short), or non-complementary oligos.

3. Generate ready-to-order CRISPR array oligos or sequence from gRNAs:
    a. Objective: The user wants to compile up to 9 pre-chosen gRNAs into ready-to-order gRNA oligos or a ready-to-order array sequence.
    b. User-system interaction: The user will use the python environment of their choice, install CRISPR_Array_Generator using pip, and then call the getArrayFromgRNA function. Here the user will input up to 9 gRNAs as python strings (one per gRNA) and run the function. The output will inform the user if any common errors are present such as non-DNA bases, CRISPR cut sites within the gRNA, or length errors (too long or too short). If no major errors are present, it will process the inputted gRNAs and output ready-to-order oligos (one forward and reverse for each gRNA input) as strings that can be annealed and cloned into a multiplexed array in the lab. It will also output an assembled DNA version of the array as a string if the user would like to skip the oligo annealing step and order the full array.

4. Generate ready-to-order CRISPR array oligos from a DNA sequence:
    a. Objective: The user wants to generate ready-to-order gRNA oligos by inputting a DNA sequence they would like to target.
    b. User-system interaction: The user will use the python environment of their choice, install CRISPR_Array_Generator using pip, and then call the getArrayFromDNA function. Here the user will input their DNA sequence and the number of gRNAs they'd like to target to the sequence of interest. The function will verify that the DNA sequence is valid, then locate all potential gRNAs present in the sequence and choose the best targets based on range targeted, likelihood of secondary structures, etc. It will inform the user of how many gRNAs were found and chosen. If suitable gRNAs are located, it will process the chosen gRNAs into ready-to-order oligos (one forward and reverse for each gRNA) as strings that can be annealed and cloned into a multiplexed array in the lab. It will also output an assembled DNA version of the array as a string if the user would like to skip the oligo annealing step and order the full array.