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 guide (g)RNA 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 proficient with excel and 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 listed in an excel file:
    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 upload an excel file listing their gRNAs to their python environment. The excel file may include as many gRNAs as the user would like to check. They will then call the check_grna function using “Array.check_grna” and input the name of their excel file without the “.xlsx”. The output will provide the user with a new excel file “grnacheck.xlsx” that lists all valid gRNA inputs and informs the user if any common errors are present such as Cas12 cut sites within the gRNA or length errors (too long or too short).

2. Check existing CRISPR gRNAs for common errors listed in an array:
    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 create an array listing all gRNAs

they would like checked. They will then call the check_grna function using "Array.check_grna" and input the name of their array". The output will provide the user with a new excel file "grnacheck.xlsx" that lists all valid gRNA inputs and informs the user if any common errors are present such as Cas12 cut sites within the gRNA or length errors (too long or too short).

3. Generate ready-to-order CRISPR array oligos or sequence from gRNAs listed in an excel file:
    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 upload an excel file listing up to 9 gRNAs they would like included in the array to their python environment. The user will then call the get_array function using "Array.get_array" and input the name of their excel file without the ".xlsx". The output will provide the user with a new excel file "grnacheck.xlsx." The first sheet will list all valid gRNA inputs and inform the user if any common errors are present such as Cas12 cut sites within the gRNA or length errors (too long or too short). If no major errors are present, the second sheet will list ready-to-order oligos (one forward and reverse for each gRNA input) that can be annealed and cloned into a multiplexed array in the lab. The sheet will also list the full sequence of the array in case the user would like to skip the oligo annealing step and order the full array.

4. Generate ready-to-order CRISPR array oligos or sequence from gRNAs listed in an array.
    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 create an array listing up to 9 gRNAs they would like to be inserted into a CRISPR array. The user will then call the get_array function using "Array.get_array" and input the name of their excel file without the ".xlsx". The output will provide the user with a new excel file "grnacheck.xlsx." The first sheet will list all valid gRNA inputs and inform the user if any common errors are present such as Cas12 cut sites within the gRNA or length errors (too long or too short). If no major errors are present, the second sheet will list ready-to-order oligos (one forward and reverse for each gRNA input) that can be annealed and cloned into a multiplexed array in the lab. The sheet will also list the full sequence of the array in case the user would like to skip the oligo annealing step and order the full array.