

Component Specification for crispr_array_generator

Willow Chernoske ; last updated 12/14/2023

Software Components

Class

Array: A set of python functions necessary to check CRISPR Cas12 guide (g)RNAs and build CRISPR arrays from gRNAs.

Functions

extract_excel_data(excel_file: str {excel sheet title without .xlsx})

- Extracts all data from an excel file and isolates the DNA components into a separate array.
- Input: The name of an excel file as a string.
- Output: All DNA components listed in the input excel file compiled into an array.

get_reverse_complement(dna: str)

- Generates the reverse complement of an input DNA sequence.
- Input: DNA sequence as a string.
- Output: Reverse complement of the input DNA sequence as a string.

make_columns_best_fit(excel_file: str {excel sheet title without .xlsx})

- Formats columns in excel files to the width of the text in each cell.
- Input: The name of an excel file as a string.
- Output: The input excel file returned with best fit column widths.

check_grna(grnas: str {excel sheet title without .xlsx} OR array)

- Checks if a gRNA has any errors (too long, too short) and removes preceding TTC PAM sequence if present.
- Inputs: The name of an excel file as a string listing gRNAs, or an array with listed gRNAs.
- Output: An excel file named “array_report.xlsx” listing all processed gRNAs and any errors.

get_array(grnas: str {excel sheet title without .xlsx} OR array)

- Generates ready-to-order forward and reverse oligonucleotides and the full array sequence of an array for up to 9 gRNAs listed in an excel file.
- Inputs: The name of an excel file as a string listing gRNAs, or an array with listed gRNAs.

- Output: An excel file named “array_report.xlsx” listing all processed gRNAs and any errors on the first sheet “gRNA check” and the designed array including individual forward and reverse oligonucleotides on the second sheet “Array”.

Interactions to Accomplish Use Cases

Use Case: Generate ready-to-order CRISPR array oligos or sequence from gRNAs submitted as an excel file.

For this use case the Array.get_array function will be used. This function will first call the check_array function, which will determine if the input is a string (excel file) or an array listing the gRNAs. If the input is a string, the function will call the extract_excel_data function to extract all the DNA components listed in the excel sheet into an array. Then, it will check for sequence or length errors. Individual processed gRNAs and any errors will be listed on the first sheet “gRNA check” of the output excel file “array_report.xlsx.” Then get_reverse_complement will be called to produce the reverse complement sequence of each gRNA. The number of input gRNAs will be counted to determine how to assemble the array oligos. Array oligos will be assembled and compiled into the second sheet “Array” of the output excel file “array_report.xlsx”. Finally, “array_report.xlsx” will go through the make_columns_best_fit function for formatting and be returned to the user. This output format is helpful because many companies accept excel workbooks when ordering plates of oligos.

Preliminary Plan (from 11/06/23)

Current status: I have working function prototypes for the checkgRNA, getReverseCompliment, and getArrayFromgRNA functions with string inputs.

Plan

1. Implement excel interface to check_grna function.
2. Import a test function to my github repository and verify that I can install it with pip.
3. Implement an excel interface to the get_array function.
4. Build a get_array_from_dna function if time allows.
5. Import all functions to my github repository and again, verify pip install works.
6. Final touches and optimizations as necessary/time allows.