

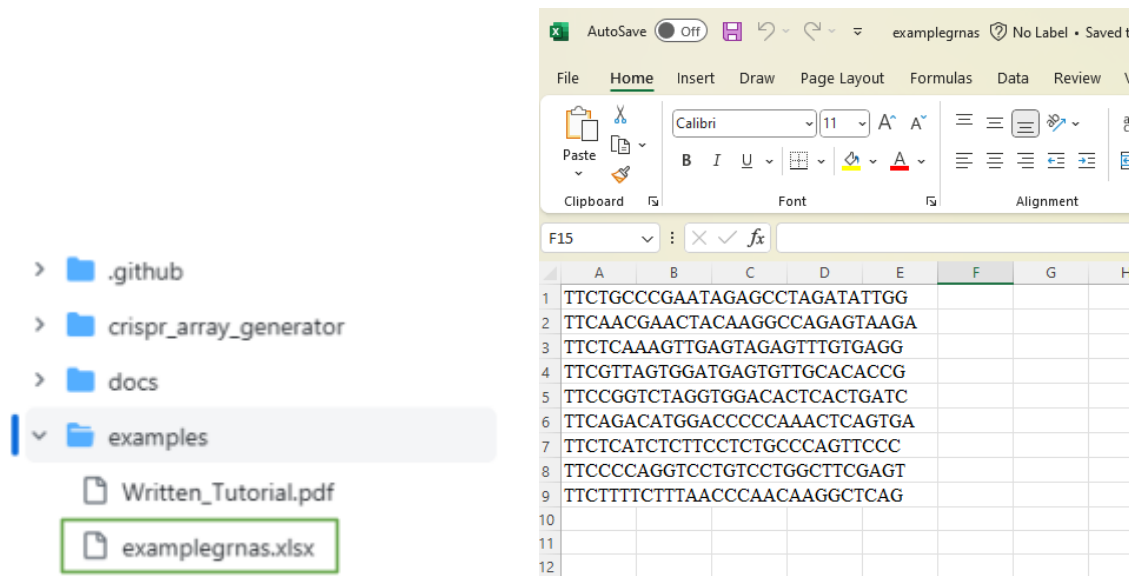
Written Tutorial for crispr_array_generator

Willow Chernoske ; last updated 12/15/2023

This is a written tutorial for generating CRISPR arrays from guide (g)RNAs using the python package `crispr_array_generator`. Feel free to follow along using the provided “examplegrnas.xlsx” excel file.

1. Obtain gRNAs

First, you must obtain the gRNAs you would like added to your array. If you are following along with the “examplegrnas.xlsx” file provided, simply download it to your computer. Otherwise, you can use a site like CHOPCHOP to obtain gRNAs and either insert them into an excel file, one guide per cell, or make a note of them elsewhere. It does not matter whether you include the TTC PAM site in front of your guide or not, as the program will automatically process these out if present. Up to 9 guides can be used to build a single array.



2. Download the crispr_array_generator using pip install

Open your python environment of choice (this tutorial uses google colab) and install `openpyxl` and `crispr_array_generator`. Then import the `Array` class from `crispr_array_generator.crisprarraygenerator` as shown below.

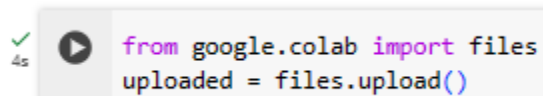
```
pip install -U openpyxl
```

```
pip install crispr_array_generator
```

```
from crispr_array_generator.crisprarraygenerator import Array
```

3. Import your gRNAs as an excel file to your python environment or add them to an array

To add a file to google colab, you can run the following code. This should prompt you to choose a file from your computer to upload.

A screenshot of a Google Colab code cell. On the left, there is a green checkmark and a play button icon. Below the play button is a small '4s' indicating execution time. The code in the cell is:

```
from google.colab import files
uploaded = files.upload()
```

```
from google.colab import files
uploaded = files.upload()
```

If you would like to add your guides to an array, you may create a python array of your chosen gRNAs, like shown.

```
grnas = ['AAAAA', 'TTTTT', 'CCCCC']
```

4. Run get_array

You're ready to run the get_array function! To do this, type "Array.get_array(**your file or array name**)" into your python environment. If using an excel sheet, add the name as a string.

```
Array.get_array('examplegrnas')
```

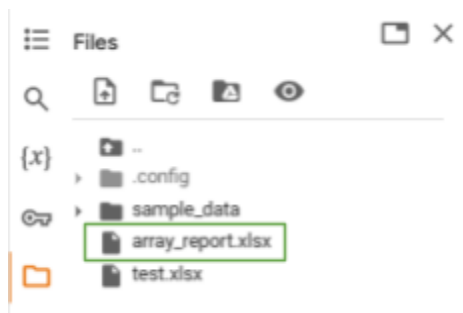
If using an array input, add the name of the array with no quotation marks.

```
Array.get_array(grnas)
```

Run the code!

5. Download and analyze array report

After running `get_array`, the program should automatically download a file “array_report.xlsx” to your python environment. In google colaboratory, this is accessible in the files tab in the top left corner.



After opening the “array_report.xlsx” file, there should be two tabs. The first “gRNA check” checks your guides for any length errors, showing an X if any are present. If any of your guides included non-nucleotide characters, they will not be included. The second sheet “Array” lists a full forward and reverse CRISPR array comprised of your input guides, as well as all forward and reverse oligonucleotides needed to build this array in the lab. You may need to scroll to see the full picture. If you try to upload more than 9 guides, you will get an error in the top right of the “Array” sheet.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	gRNAs		length error (>24 nucleotides)	length error (<20 nucleotides)									
2	TGCCGAATAGAGCTAGATTGG	X											
3	AACGAACTACAAGGCCAGTAAGA	X											
4	TCAAAGTTGAGTAGAGTTGTGAGG	X											
5	GTTAGTGGATGAGTTGCACACCG	X											
6	CGGTCTAGGTGGACACTCACTGATC	X											
7	AGACATGGACCCCAAACTCACTGA	X											
8	TCATCTCTCTCTGCCCAGTTCCC	X											
9	CCCAGGTCTGTCTGCTTCGAGT	X											
10	TTTTCTTAACCCCAAGGCTCAG	X											
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													

	A
1	Full array fwd:
2	CCCTAAATAATTTCTACTGTTGTAGATTGCCGAATAGAGCTAGATATTGGTGCAATAATTTCTACTGTTGTAGATAACGAAGCCAGAGTAAGATTCTAAATAATTTCTACTGTTGTAGATTCAAAGTTGAGTAGAGTTGTGAGGCA
3	
4	gRNA #: Fwd oligos:
5	1 CCCTAAATAATTTCTACTGTTGTAGATTGCCGAATAGAGCTAGATATTGG
6	2 TGCGAAATAATTTCTACTGTTGTAGATAACGAAGCCAGAGTAAGATTCT
7	3 AAATAATTTCTACTGTTGTAGATTCAAAGTTGAGTAGAGTTGTGAGG
8	4 CAATAAATAATTTCTACTGTTGTAGATGTTAGTGGATGAGTTGACACCGTATG
9	5 AAATAATTTCTACTGTTGTAGATCGGTCTAGGTGGACACTCACTGATC
10	6 AGAAAAATAATTTCTACTGTTGTAGATGACATGGACCCCAAACTCACTGATACA
11	7 AAATAATTTCTACTGTTGTAGATTCACTCTCTCTGCCCAGTTCCC
12	8 GCTGAAATAATTTCTACTGTTGTAGATCCAGGTCTGTCTGGCTTCGAGTGAGT
13	9 CGTCTGAGCCTGTTGGGTTAAAGAAAAATCTACACAGTAGAAATTATTTACTC
14	
15	
16	
17	
18	
19	
20	
21	