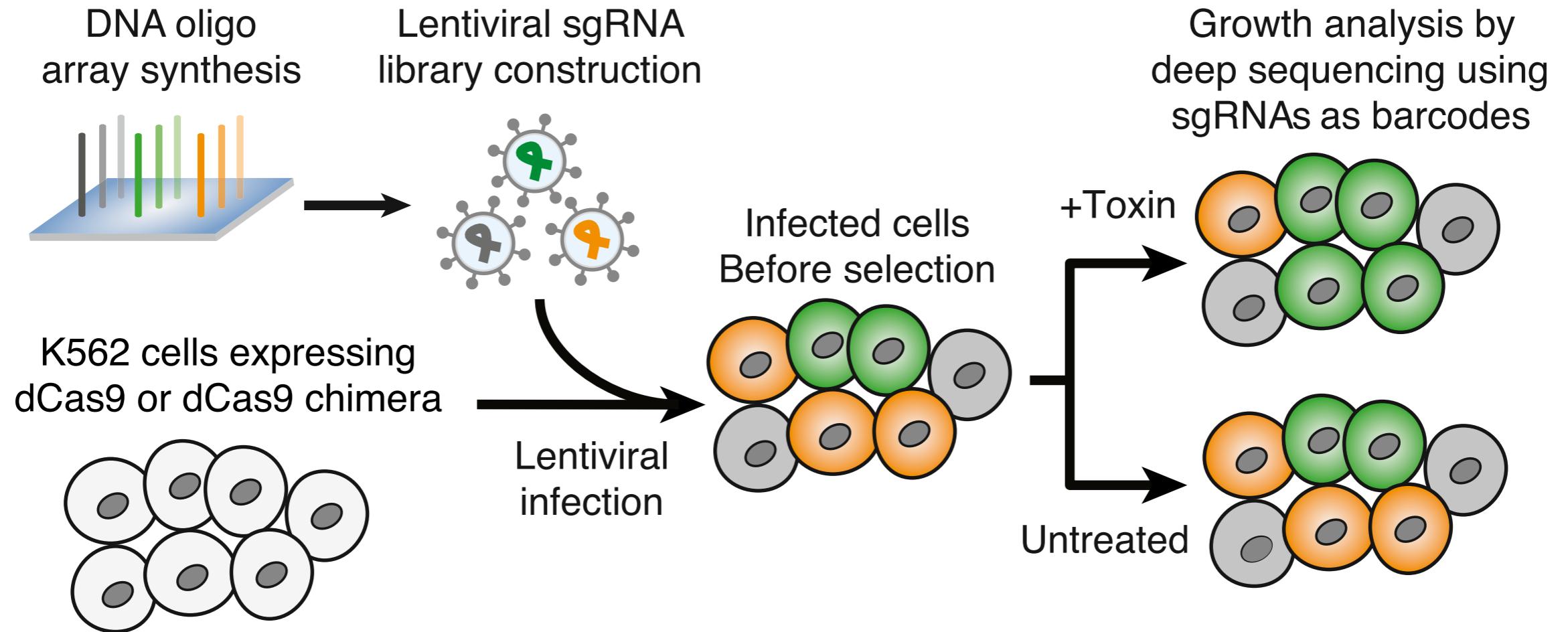


Outline

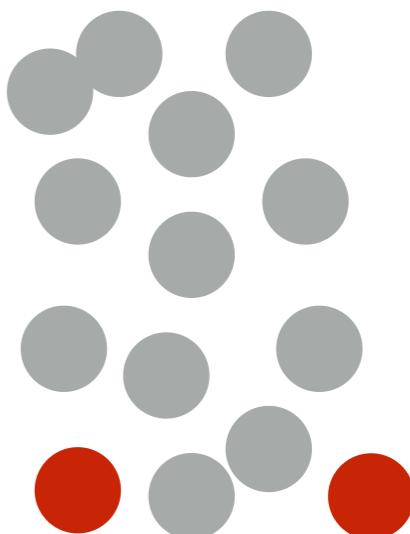
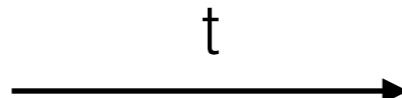
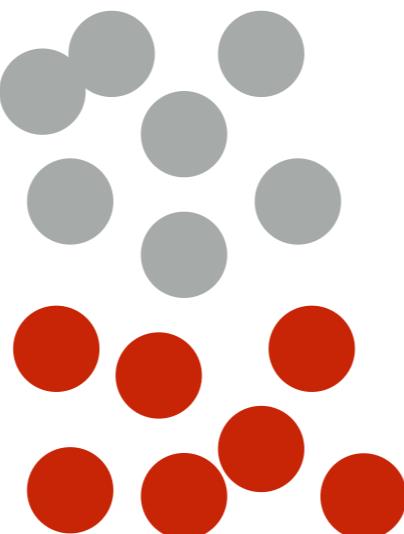
- Brief review of pooled screening
- Download and install software
- Step 1: counting sgRNAs in raw sequencing files
- Step 2: calculating sgRNA-level and gene-level phenotypes and p-values
- Interpreting key graphs
- Step 3: making custom graphs
- Downstream analysis with GO and StringDB



Common types of screens

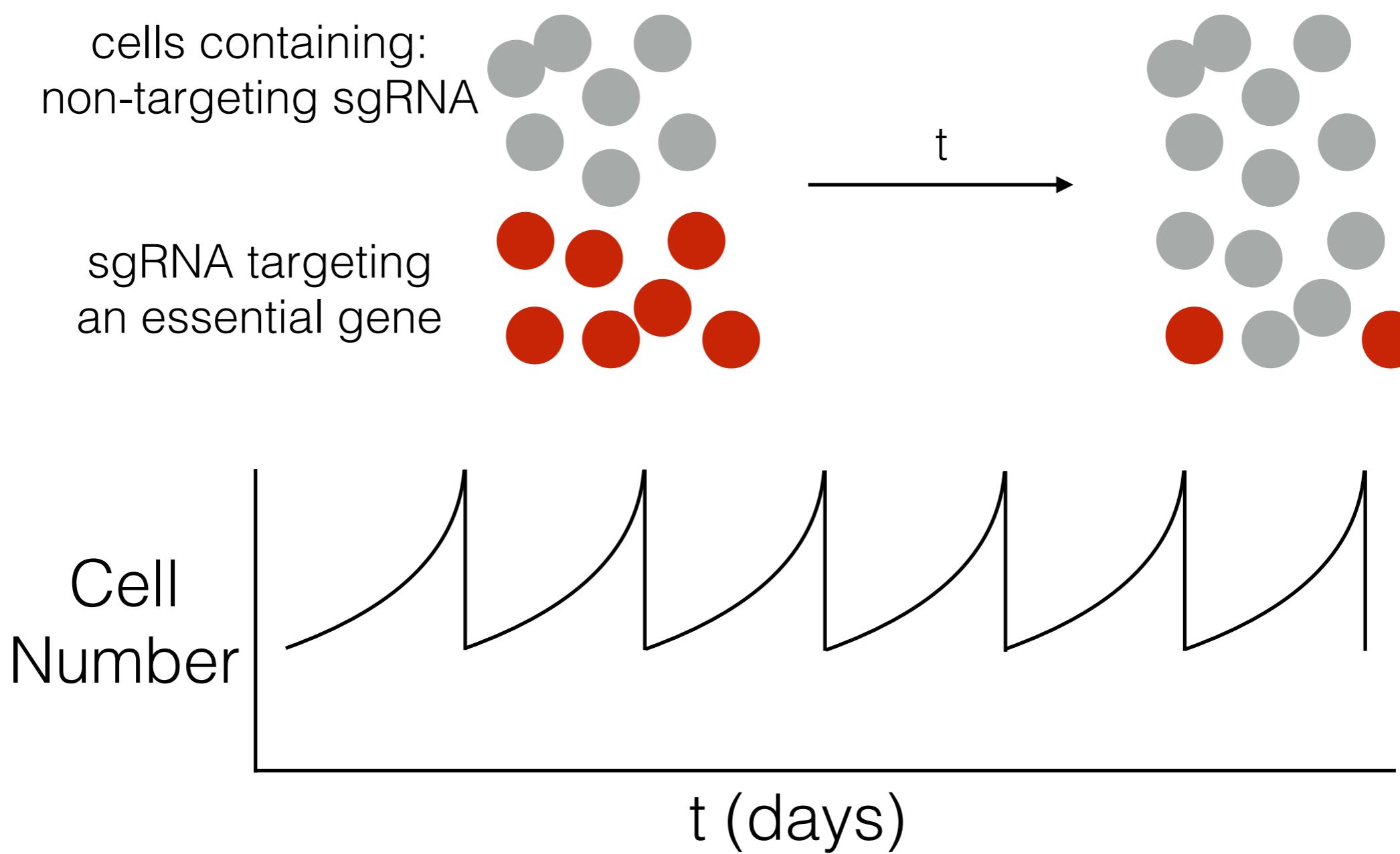
- Growth based

cells containing:
non-targeting sgRNA



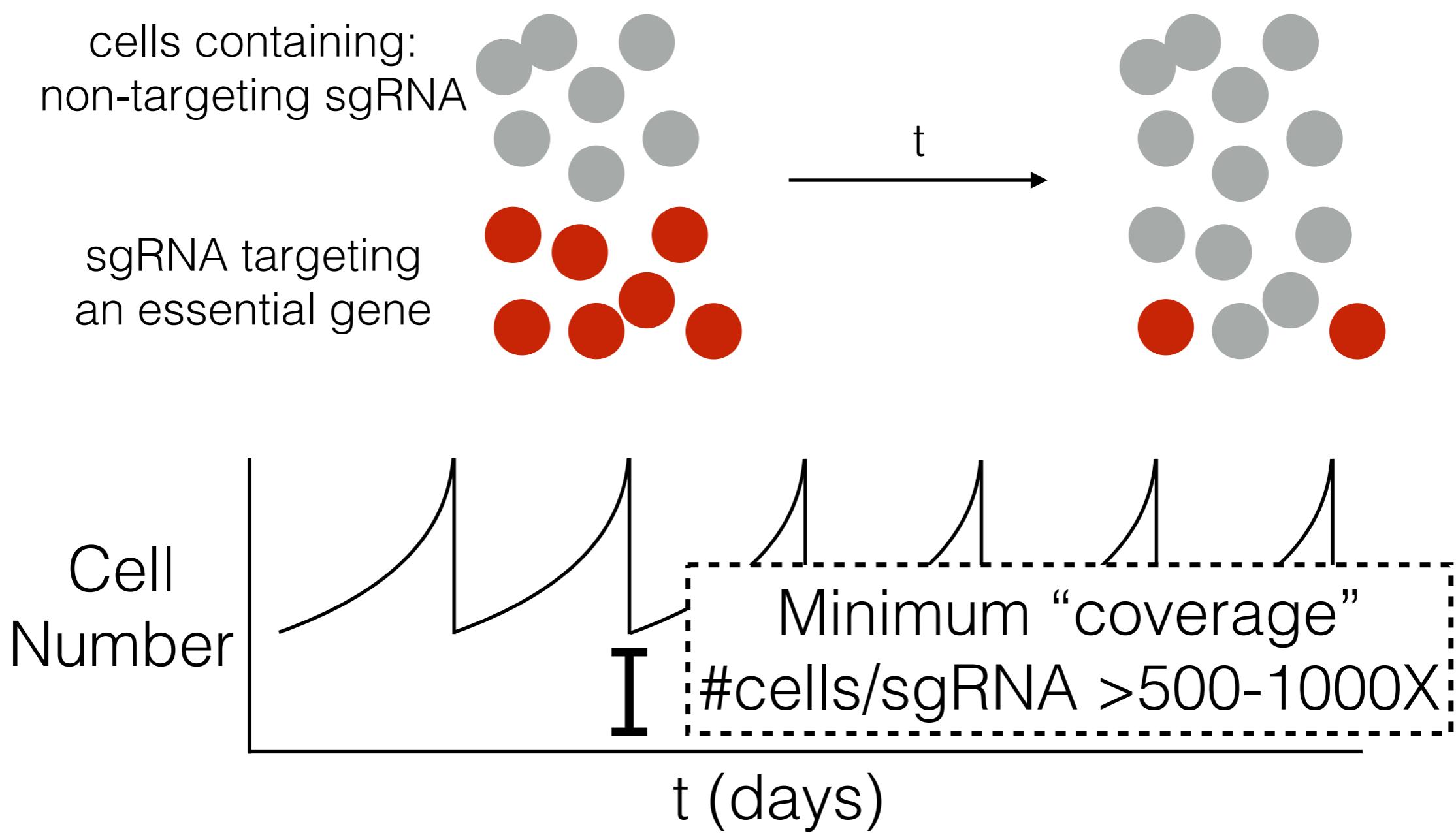
Common types of screens

- Growth based



Common types of screens

- Growth based

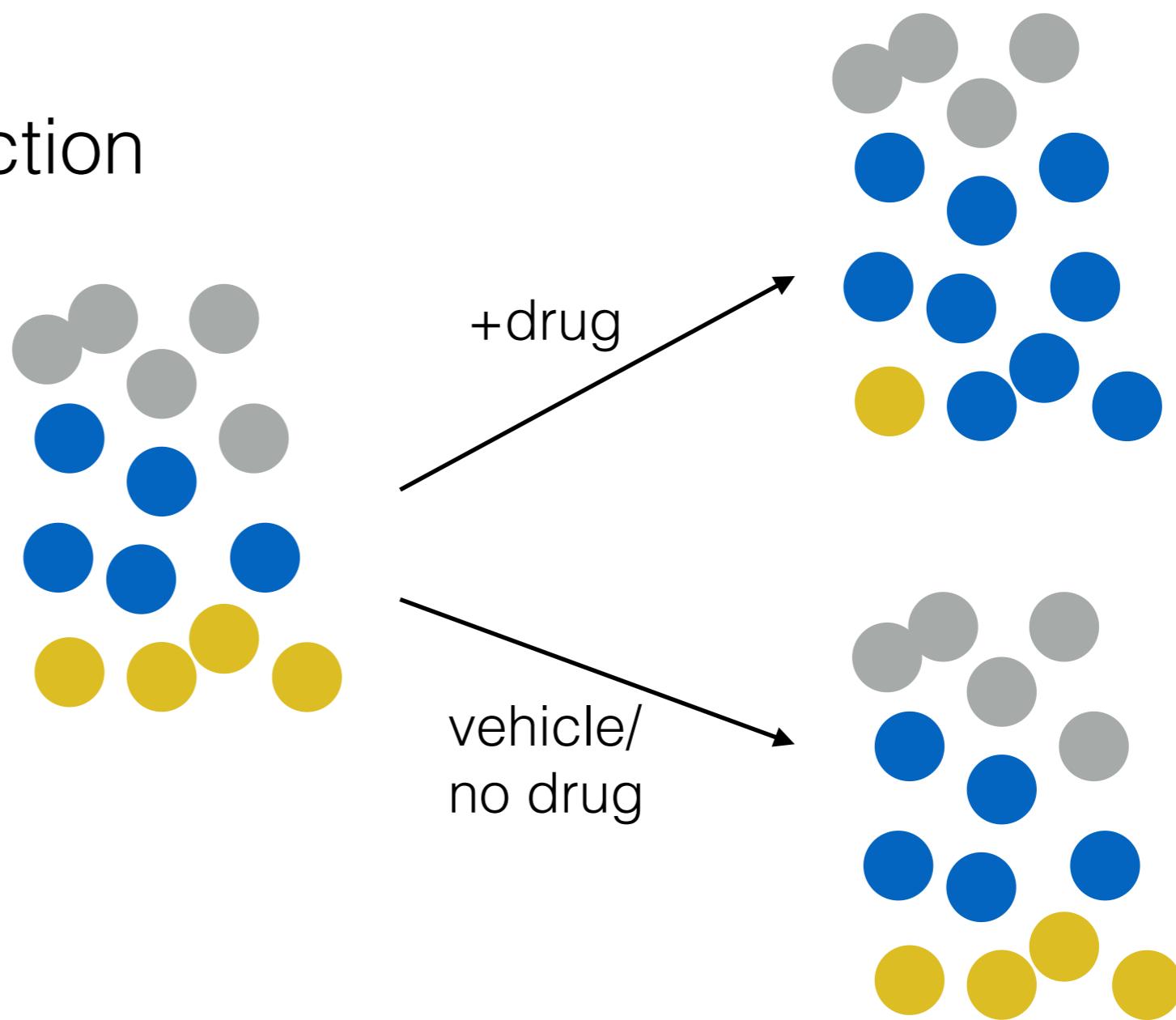


Common types of screens

- Growth based
- Drug/Toxin selection

cells containing:
non-targeting sgRNA
sgRNA that promotes
resistance

sensitivity

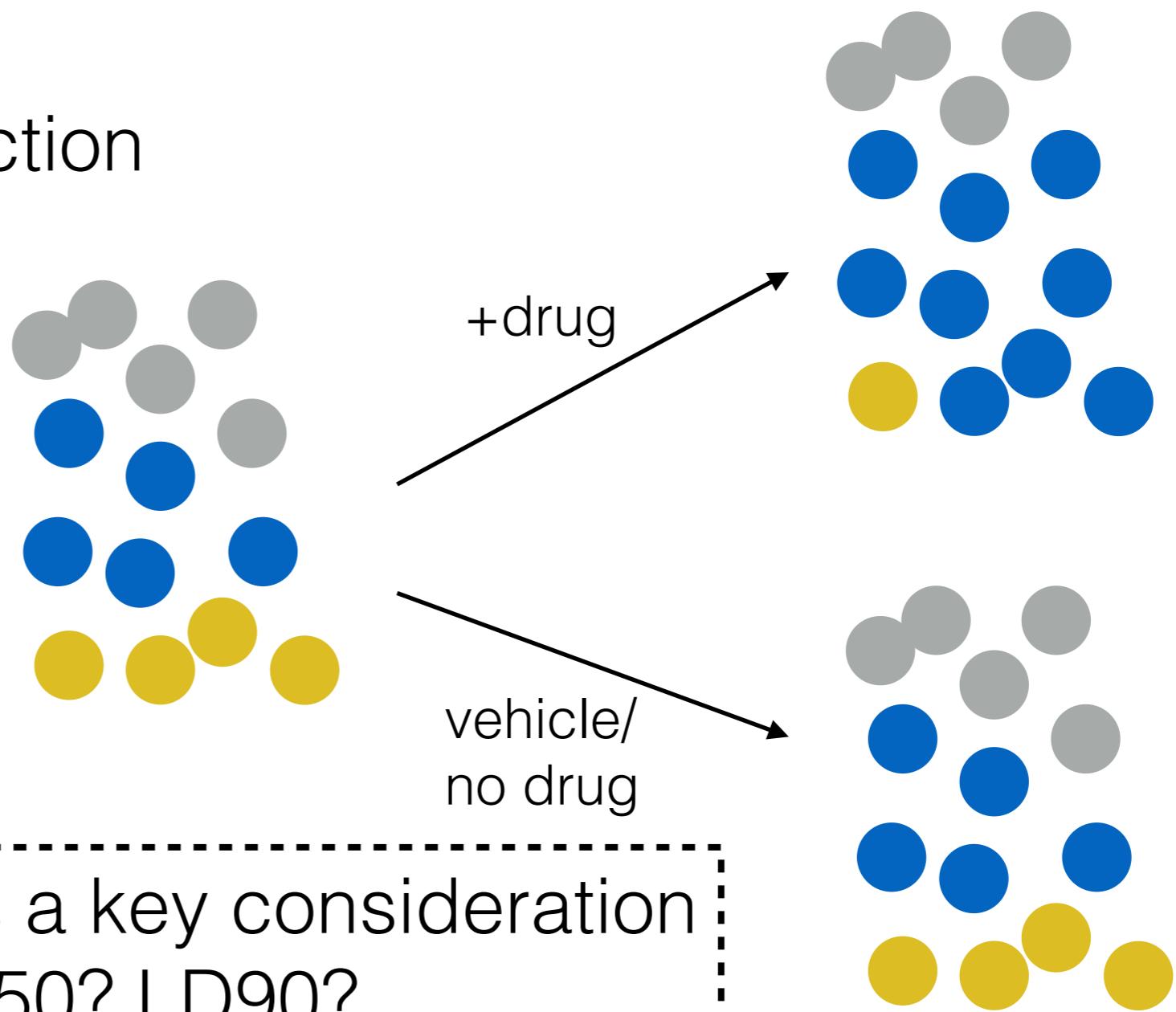


Common types of screens

- Growth based
- Drug/Toxin selection

cells containing:
non-targeting sgRNA
sgRNA that promotes
resistance

sensitivity

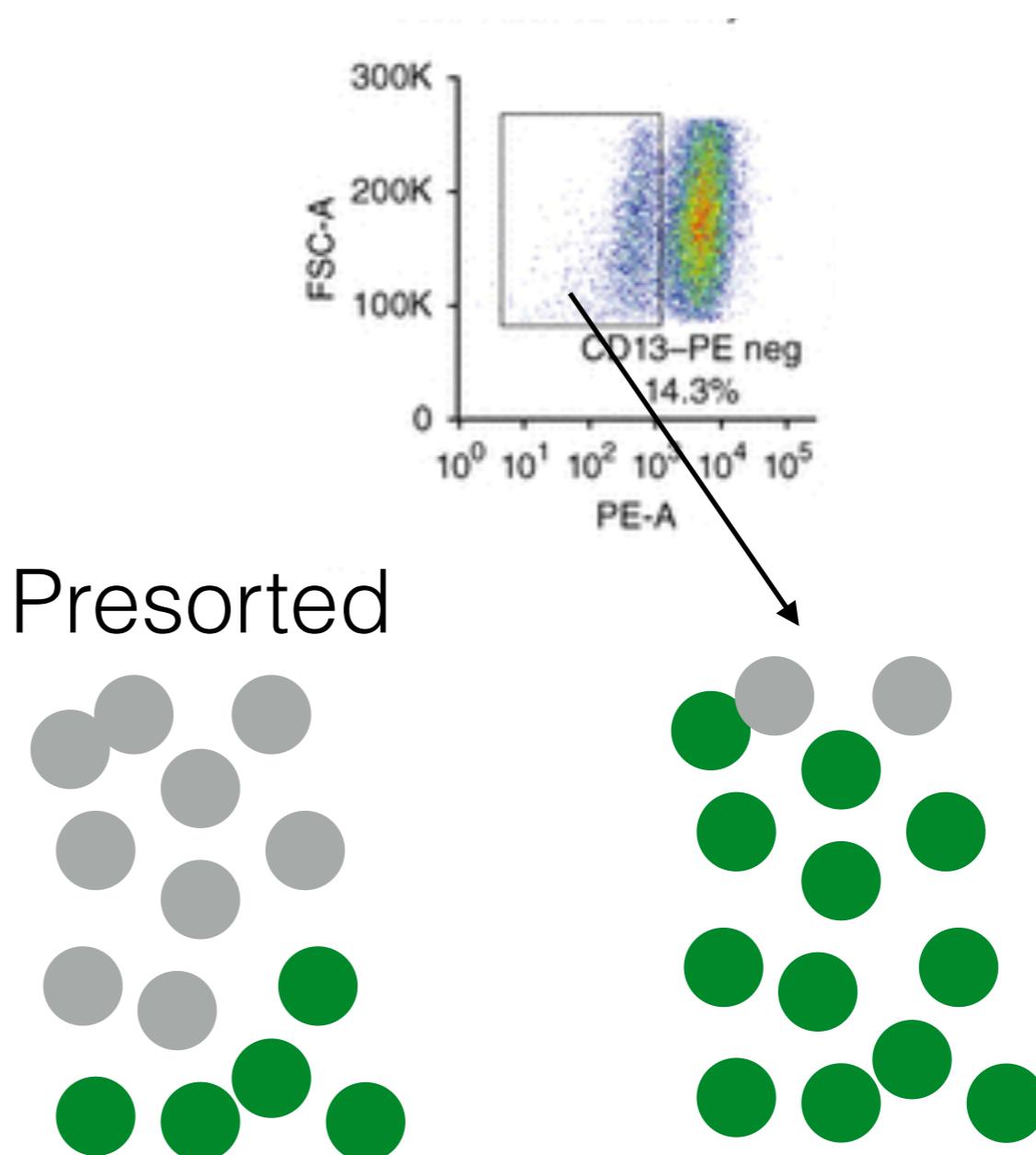


Minimum coverage is a key consideration
Dosing: LD50? LD90?

Common types of screens

- Growth based
- Drug/Toxin selection
- FACS sorting

cells containing:
non-targeting sgRNA
sgRNAs targeting CD13



How many cells do you
need to grow for a screen?

How many cells do you need to grow for a screen?

#genes
#sgRNAs/gene
#cells/sgRNA

multiple conditions
replicates

How many cells do you need to grow for a screen?

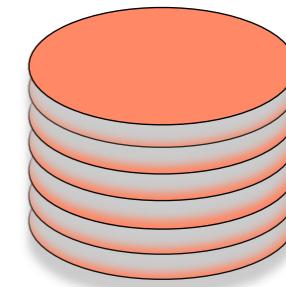
#genes	
#sgRNAs/gene	
#cells/sgRNA	
multiple conditions	
replicates	

20,000	
x10	
<u>x1000</u>	
200M cells	
at minimum	

How many cells do you need to grow for a screen?

	#genes
	#sgRNAs/gene
	#cells/sgRNA
multiple conditions replicates	

20,000
x10
x1000
200M cells
at minimum

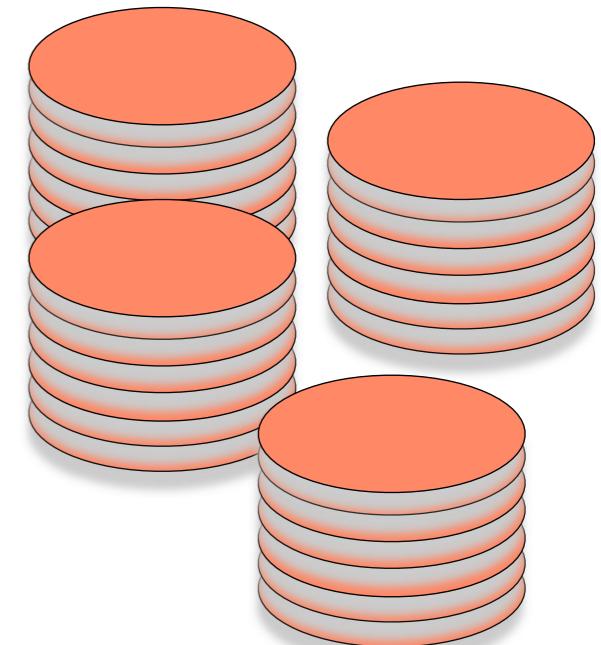


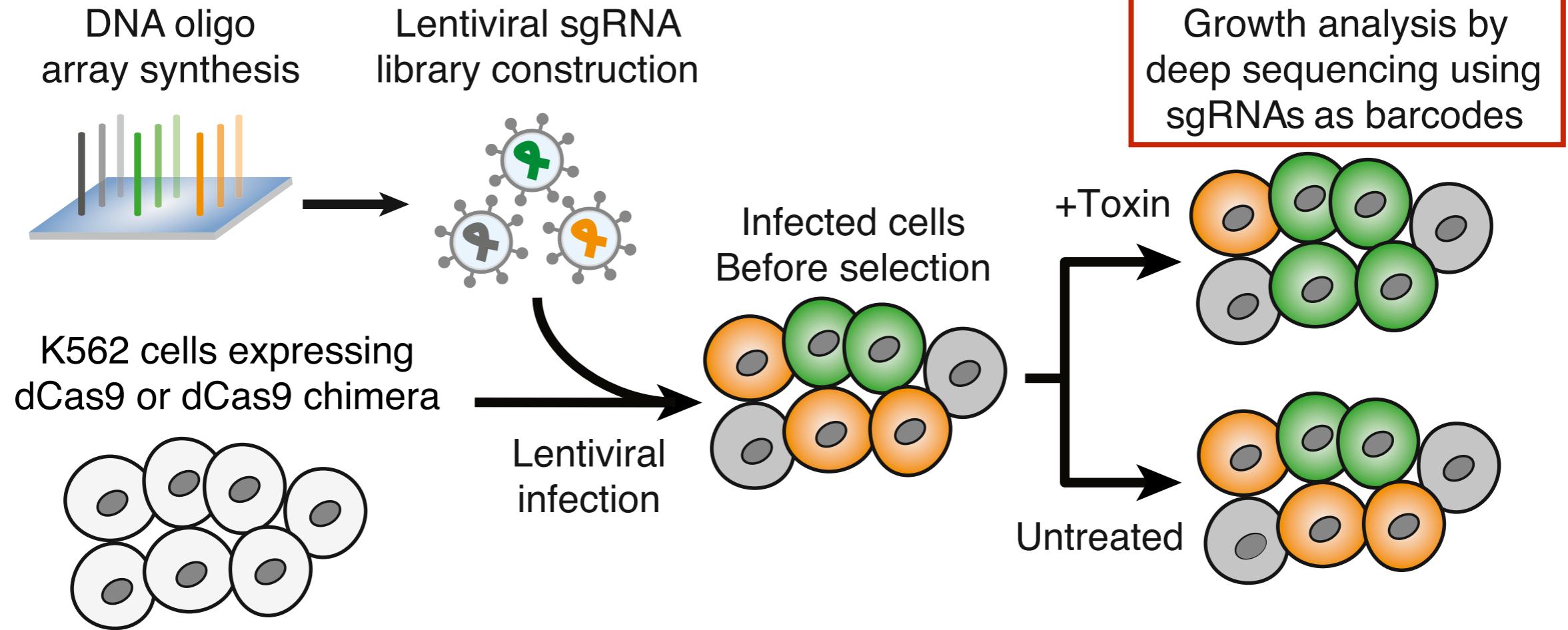
How many cells do you need to grow for a screen?

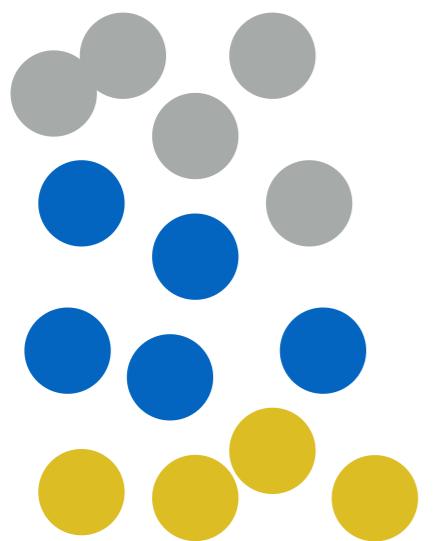
#genes
#sgRNAs/gene
#cells/sgRNA

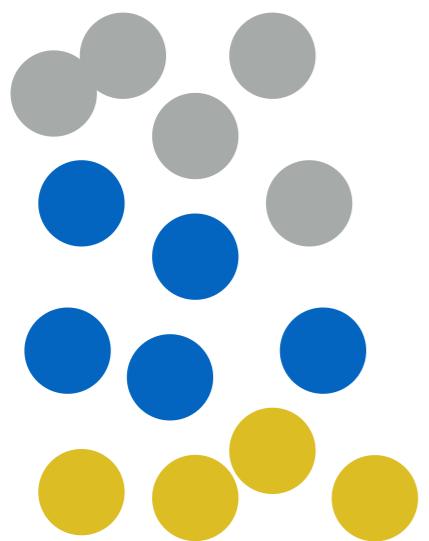
multiple conditions
replicates

20,000
x10
x1000
200M cells
at minimum

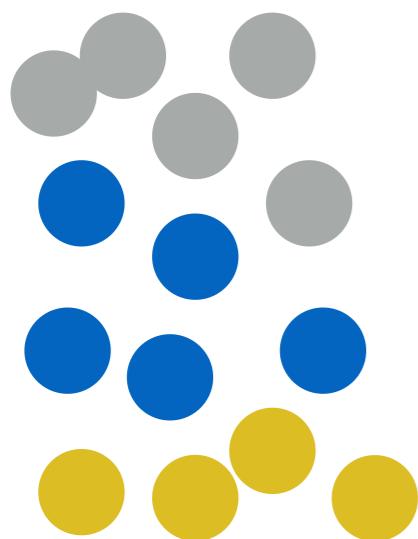




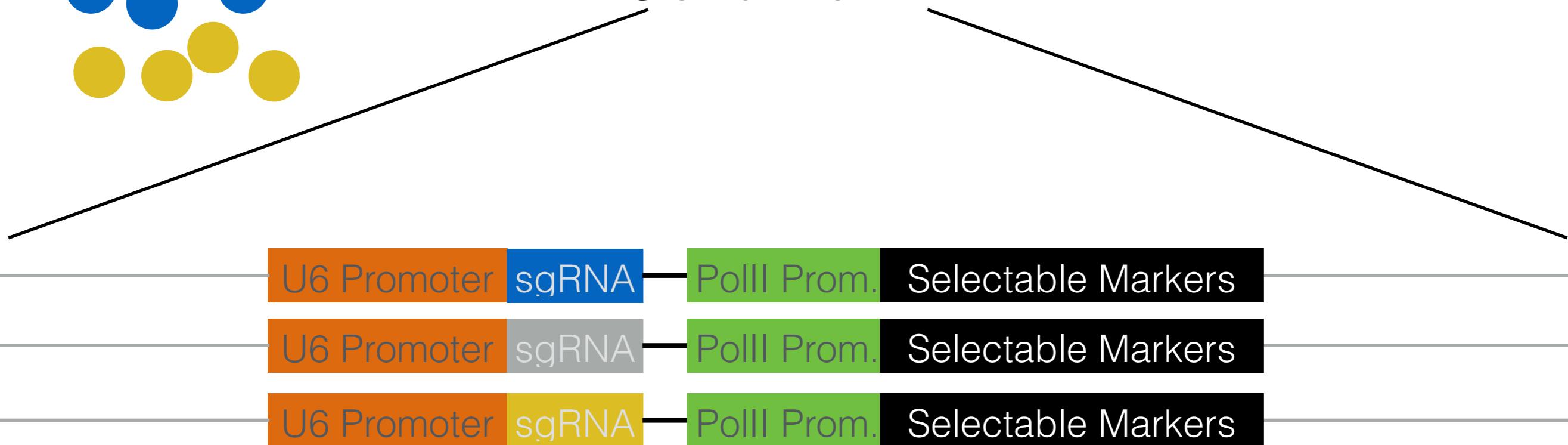


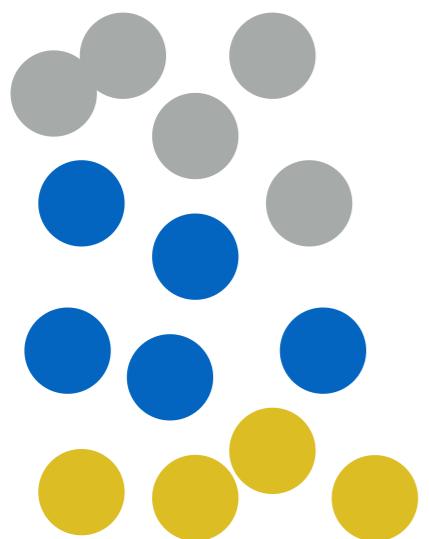


Pool of
Genomic DNA



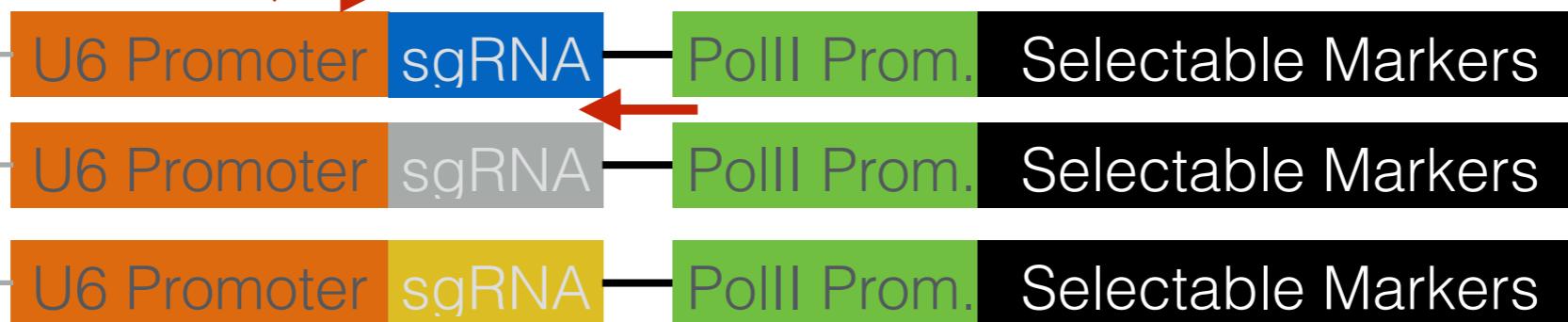
Pool of
Genomic DNA

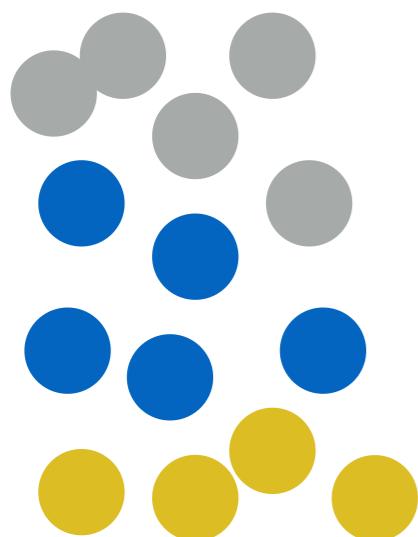




Pool of
Genomic DNA

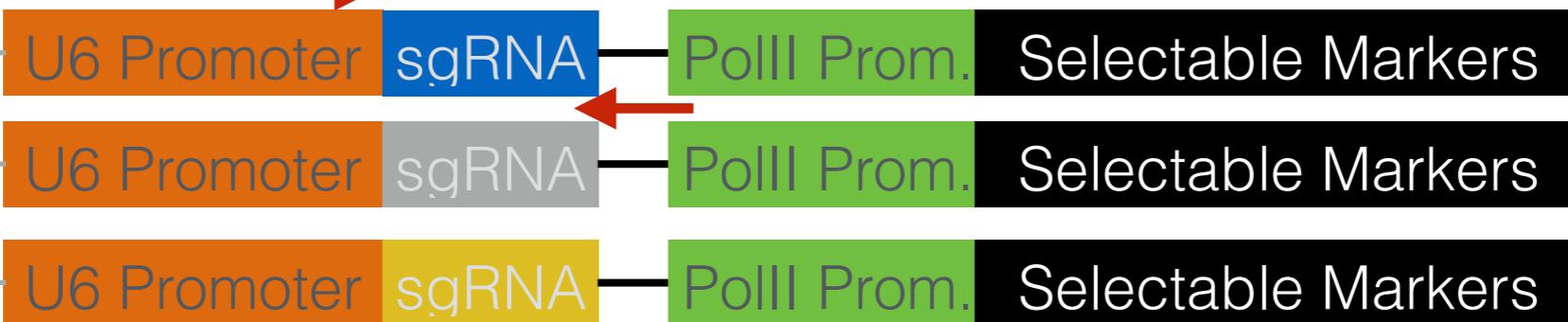
illumina adapters



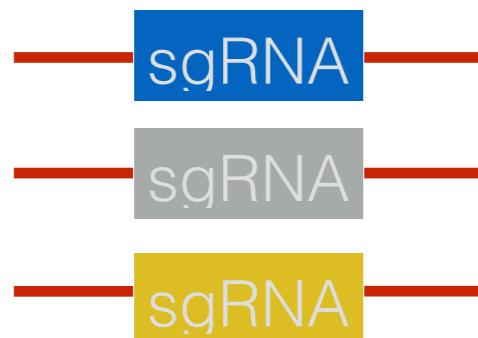


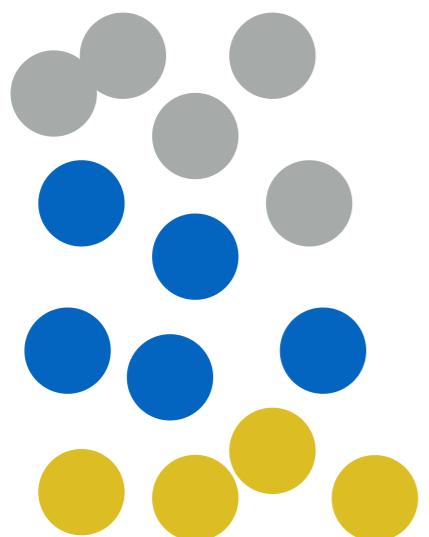
Pool of Genomic DNA

illumina adapters



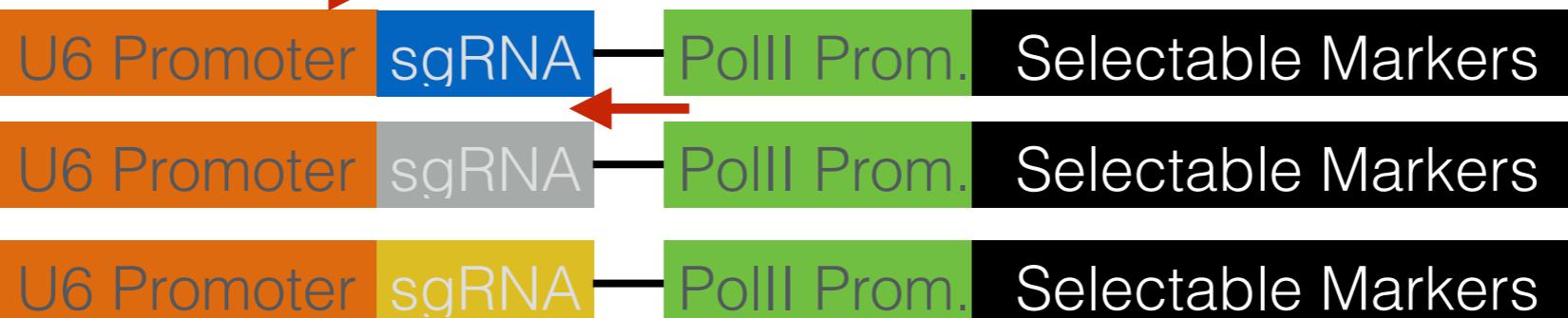
PCR





Pool of Genomic DNA

illumina adapters



PCR



Next-Gen Sequencing
Align back to the library
Count

		Untreated	Treated
sgGeneB	GACTGCCGCCTAGAACATGAGG	500	1000
sgGeneY	GGGTTGGACCATCTTCTTA	750	100
negCtrl	GCCACCTGCCAAAGACACTC	550	575
...			

Setup!

- Install python 2.7, along with ipython, numpy, scipy, pandas, matplotlib packages
- Alternately, download and install Canopy
 - <https://store.enthought.com/downloads/>



Enthought Canopy: Easy Python Deployment Plus Integrated Analysis Environment for Scientific Computing, Data Analysis and Engineering

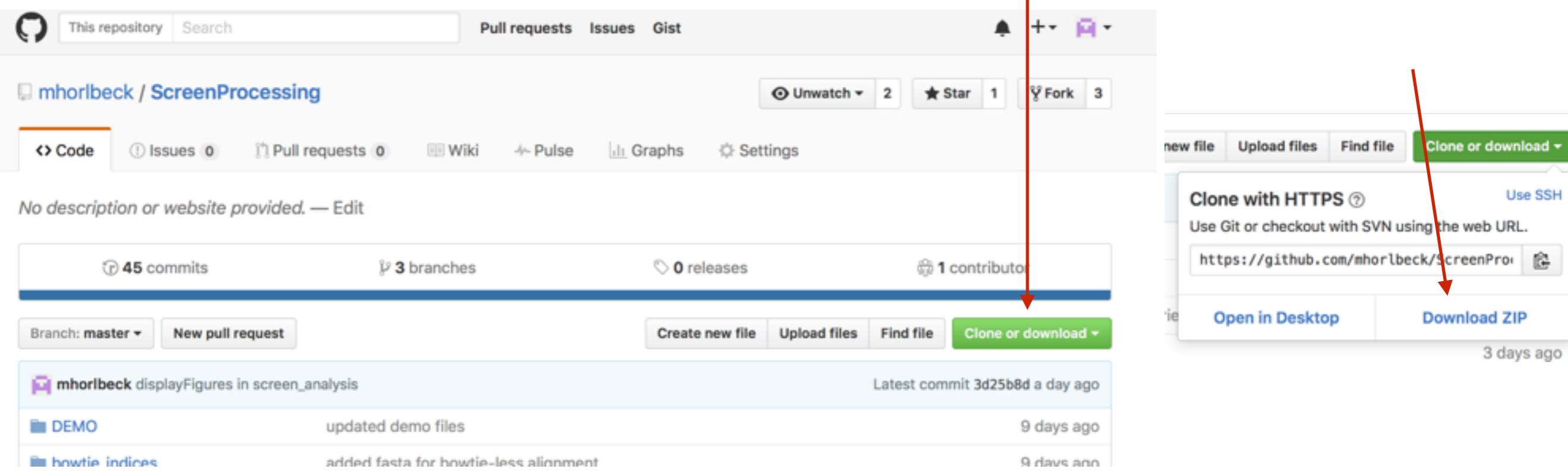
FREE for all users: Canopy Express, which includes access to 100+ of Canopy's most popular Python packages for scientific computing, data analysis, and engineering PLUS Canopy's integrated analysis environment. Get started today with easy deployment of pre-built, tested, and dependency-aware packages such as NumPy, SciPy, Pandas, Matplotlib, IPython and more.

DOWNLOAD
Canopy

By downloading Canopy you acknowledge your acceptance of all the terms and conditions of the applicable license.

Setup!

- Download analysis pipeline scripts
 - <https://github.com/mhorlbeck/ScreenProcessing>
 - download as zip and unzip on your computer, keeping track of the folder where you unzipped the files



File paths

- You always have a current working directory

- Identify yours with **pwd**

```
[In [1]: pwd  
Out[1]: u'/home/mhorlbeck'
```

- You can change the working directory with **cd**

```
[In [2]: cd home_tmp/mhorlbeck/ScreenProcessing/  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing
```

```
[In [3]: pwd  
Out[3]: u'/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing'
```

- There are “absolute paths” starting with “/“ or “C://“ and “relative paths” relative to your working directory
- There are several additional shortcuts you can use:
 - . = current directory
 - .. = parent directory
 - ~ = home directory

Running scripts

- Start by **cd**-ing into the unzipped ScreenProcessing directory

```
[In 2]: cd home_tmp/mhorlbeck/ScreenProcessing/  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing
```

```
[In 3]: pwd  
Out[3]: u'/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing'
```

- Run scripts using the **run** command—try running the first script we'll use with

run fastqgz_to_counts.py -h

```
[In 5]: run fastqgz_to_counts.py -h  
usage: fastqgz_to_counts.py [-h] [-p PROCESSORS] [--trim_start TRIM_START]  
                           [--trim_end TRIM_END] [--test]  
                           Library_Fasta Out_File_Path Seq_File_Names  
                           [Seq_File_Names ...]  
  
Process raw sequencing data from screens to counts files in parallel  
  
positional arguments:  
  Library_Fasta      Fasta file of expected library reads.  
  Out_File_Path      Directory where output files should be written.  
  Seq_File_Names     Name(s) of sequencing file(s). Unix wildcards can be  
                    used to select multiple files at once. The script will  
                    search for all *.fastq.gz, *.fastq, and  
                    *.fa(/fasta/fna) files with the given wildcard name.  
  
optional arguments:  
  -h, --help           show this help message and exit  
  -p PROCESSORS, --processors PROCESSORS  
  --trim_start TRIM_START  
  --trim_end TRIM_END  
  --test              Run the entire script on only the first 10000 reads of  
                     each file. Be sure to delete or move all test files  
                     before re-running script as they will not be  
                     overwritten.
```

Step 1: fastqgz_to_counts.py

counting sgRNAs in raw sequencing files



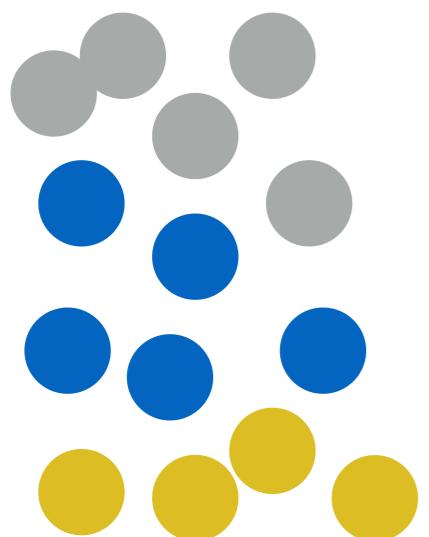
Step 2: process_experiments.py

calculating sgRNA-level and gene-level phenotypes and p-values



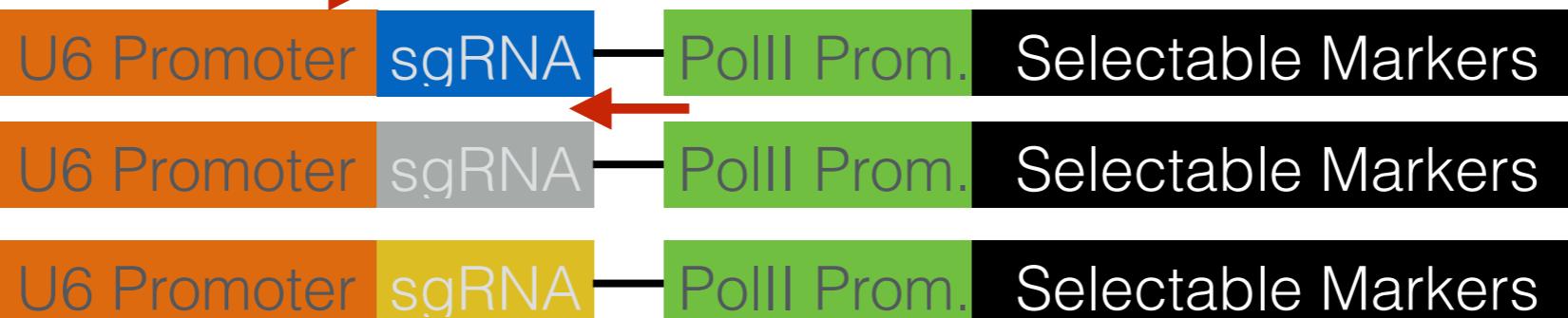
Step 3: screen_analysis.py

making custom graphs

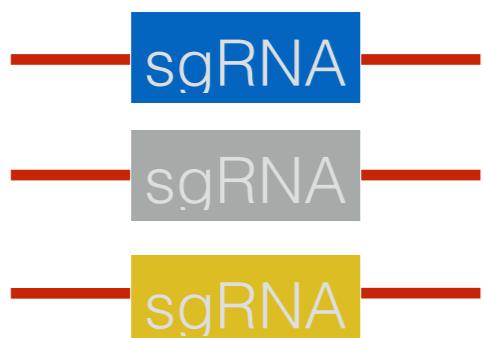


Pool of Genomic DNA

illumina adapters



PCR



Next-Gen Sequencing
Align back to the library

Count

		Untreated	Treated
sgGeneB	GAATGCCGCTAGAACATGAGG	500	1000
sgGeneY	GGGTTGGACCATCTTCTTA	750	100
negCtrl	GCCACCTGCCAAAGACACTC	550	575
...			

Raw sequencing files

- Let's take a look at the data we get back from the sequencer—look in Demo/Step1/Sequencing_files
 - open any file named _____.fastq in a text editor

Raw sequencing files

- Let's take a look at the data we get back from the sequencer—look in Demo/Step1/Sequencing_files
 - open any file named _____.fastq in a text editor

```
@HWI-ST640:851:C4U9HACXX:8:1101:1149:2177 1:N:0:ATCACG
GGGACCCAGCCTAGACGTGTGTTAAGAGCTAACGGAAACAGCATAGCA
+
@@@DBDDD3CFB?F9:C<C+ACGGII9C:EGGIDFGIGFGFFIBGFIIII
@HWI-ST640:851:C4U9HACXX:8:1101:1160:2199 1:N:0:ATCACG
GTTCCAGCCCCACCGCACCGCGTTAACGGCTAACGGAAACAGCATAGC
+
@@@DDFFFHHGHGIJJJJIJGHHIJJ9BGHGFJJJHBCHHHI@GGIGCE
@HWI-ST640:851:C4U9HACXX:8:1101:1225:2207 1:N:0:ATCACG
GCCTGTGACTTAAGGGGCAGTTAACGGCTAACGGAAACAGCATAGCA
+
• RRRFFFFFHGGT1111T11TT1T1AHCNHTDHGGCFHH1TCN1T1111
```

Raw sequencing files

- Let's take a look at the data we get back from the sequencer—look in Demo/Step1/Sequencing_files
 - open any file named _____.fastq in a text editor

```
@HWI-ST640:851:C4U9HACXX:8:1101:1149:2177 1:N:0:ATCACG
GGGACCCAGCCTAGACGTGTGTTAAGAGCTAACGGAAACAGCATAGCA
+
@@@DBDDD3CFB?F9:C<C+ACGGII9C:EGGIDFGIGFGFFIBGFIIII
@HWI-ST640:851:C4U9HACXX:8:1101:1160:2199 1:N:0:ATCACG
GTTCCAGCCCCACCGCACCGCGTTAACGGCTAACGGAAACAGCATAGC
+
@@@DDFFFHHGHGIJJJJIJGHHIJJ9BGHGFJJJHBCHHHI@GGIGCE
@HWI-ST640:851:C4U9HACXX:8:1101:1225:2207 1:N:0:ATCACG
GCCTGTGACTTAAGGGGCAGTTAACGGCTAACGGAAACAGCATAGCA
+
• RRRFFFFFHGGT1111T11TT1T1AHCNHTDHGGCFHH1TCN1T1111
```

Raw sequencing files

- Let's take a look at the data we get back from the sequencer—look in Demo/Step1/Sequencing_files
 - open any file named _____.fastq in a text editor

```
@HWI-ST640:851:C4U9HACXX:8:1101:1149:2177 1:N:0:ATCACG  
GGGACCCAGCCTAGACGTGTGTTAAGAGCTAACGGAAACAGCATAGCA
```

sequence ID

+

```
@@@DBDDD3CFB?F9:C<C+ACGGII9C:EGGIDFGIGFGFFIBGFIIII
```

```
@HWI-ST640:851:C4U9HACXX:8:1101:1160:2199 1:N:0:ATCACG  
GTTCCAGCCCCACCGCACCGCGTTAAGAGCTAACGGAAACAGCATAGC
```

+

```
@@@DDFFFHHGHGIJJJJIJGHHIJJ9BGHGFJJJHBCHHHI@GGIGCE
```

```
@HWI-ST640:851:C4U9HACXX:8:1101:1225:2207 1:N:0:ATCACG  
GCCTGTGACTTAAGGGGCAGTTAAGAGCTAACGGAAACAGCATAGCA
```

+

```
• RRRFFFFFHGGT1111T11TT1T1AH?CHHTDHGGCFHH1TCCH1T1111
```

Raw sequencing files

- Let's take a look at the data we get back from the sequencer—look in Demo/Step1/Sequencing_files
 - open any file named _____.fastq in a text editor

```
@HWI-ST640:851:C4U9HACXX:8:1101:1149:2177 1:N:0:ATCACG  
GGGACCCAGCCTAGACGTGTGTTAAGAGCTAACGGAAACAGCATAGCA  
+  
@@@DBDDD3CFB?F9:C<C+ACGGII9C:EGGIDFGIGFGFFIBGFIIII  
@HWI-ST640:851:C4U9HACXX:8:1101:1160:2199 1:N:0:ATCACG  
GTTCCAGCCCCACCGCACCGCGTTAACGGAGCTAACGGAAACAGCATAGC  
+  
@@@DDFFFHHGHGIJJJJIJGHHIJJ9BGHGFJJJHBCHHHI@GGIGCE  
@HWI-ST640:851:C4U9HACXX:8:1101:1225:2207 1:N:0:ATCACG  
GCCTGTGACTTAAGGGGCAGTTAACGGAGCTAACGGAAACAGCATAGCA  
+  
• RRRFFFFFHGGT1111T11TT1T1AHCCHHTDHGGCEHH1TCCH1T1111
```

sequence ID
sequence read

Raw sequencing files

- Let's take a look at the data we get back from the sequencer—look in Demo/Step1/Sequencing_files
 - open any file named _____.fastq in a text editor

```
@HWI-ST640:851:C4U9HACXX:8:1101:1149:2177 1:N:0:ATCACG  
GGGACCCAGCCTAGACGTGTGTTAAGAGCTAACGGAAACAGCATAGCA
```

sequence ID
sequence read

```
+  
@@@DBDDD3CFB?F9:C<C+ACGGII9C:EGGIDFGIGFGFFIBGFIIII
```

quality scores

```
@HWI-ST640:851:C4U9HACXX:8:1101:1160:2199 1:N:0:ATCACG  
GTTCCAGCCCCACCGCACCGCGTTAACGGAGCTAACGGAAACAGCATAGC
```

```
+  
@@@DDFFFHHGHGIJJJJIJGHHIJJ9BGHGFJJJHBCHHHI@GGIGCE
```

```
@HWI-ST640:851:C4U9HACXX:8:1101:1225:2207 1:N:0:ATCACG  
GCCTGTGACTTAAGGGGCAGTTAACGGAGCTAACGGAAACAGCATAGCA
```

```
+  
•RRFFFFFHGGHGT1111T11TT1T1AHCCHHTDHGGCFHH1TCCH1T1111
```

Raw sequencing files

- Let's take a look at the data we get back from the sequencer—look in Demo/Step1/Sequencing_files
 - open any file named _____.fastq in a text editor

```
@HWI-ST640:851:C4U9HACXX:8:1101:1149:2177 1:N:0:ATCACG  
GGGACCCAGCCTAGACGTGTGTTAAGAGCTAACGGAAACAGCATAGCA  
+  
@@@DBDDD3CFB?F9:C<C+ACGGII9C:EGGIDFGIGFGFFIBGFIIII  
@HWI-ST640:851:C4U9HACXX:8:1101:1160:2199 1:N:0:ATCACG  
GTTCCAGCCCCACCGCACCGCGTTAAGAGCTAACGGAAACAGCATAGC  
+  
@@@DDFFFHHGHGIJJJJIJGHHIJJ9BGHGFJJJHBCHHHI@GGIGCE  
@HWI-ST640:851:C4U9HACXX:8:1101:1225:2207 1:N:0:ATCACG  
GCCTGTGACTTAAGGGGCAGTTAAGAGCTAACGGAAACAGCATAGCA  
+  
•RRFFFFFHGGT1111T11TT1T1AHCCHHTDHGGCEHH1TCCH1T1111
```

sequence ID
sequence read

quality scores

Raw sequencing files

- Let's take a look at the data we get back from the sequencer—look in Demo/Step1/Sequencing_files
 - open any file named _____.fastq in a text editor

```
@HWI-ST640:851:C4U9HACXX:8:1101:1149:2177 1:N:0:ATCACG  
GGGACCCAGCCTAGACGTGTGTTAAGAGCTAACGGAAACAGCATAGCA  
+  
@@@DBDDD3CFB?F9:C<C+ACGGII9C:EGGIDFGIGFGFFIBGFIIII  
@HWI-ST640:851:C4U9HACXX:8:1101:1160:2199 1:N:0:ATCACG  
GTTCCAGCCCCACCGCACCGCGTTAAGAGCTAACGGAAACAGCATAGC  
+  
@@@DDFFFHHGHGIJJJJIJGHHIJJ9BGHGFJJJHBCHHHI@GGIGCE  
@HWI-ST640:851:C4U9HACXX:8:1101:1225:2207 1:N:0:ATCACG  
GCCTGTGACTTAAGGGGCAGTTAAGAGCTAACGGAAACAGCATAGCA  
+  
•RRFFFFFHGGT1111T11TT1T1AH?CHHTDHGGCEHH1TCCH1T1111
```

sequence ID
sequence read

quality scores

library fasta file

- Can be found in /library_reference/
CRISPRi_v1_human.trim_1_35.fa

```
>Drug_Targets+Kinase_Phosphatase=CDK10_+_89753127.24-all~e39m1  
ATCTGCTCGCACTCCAGATCGTTAAGAGCTAAG  
>Drug_Targets+Kinase_Phosphatase=HRH2_--_175108555.26-all~e39m1  
CCTCAGCTTGTTCCTGCCAGTGTAAAGAGCTA  
>Apoptosis+Cancer+Other_Cancer=CUL1_+_148395795.23-ENST00000602748.1~e39m1  
GGCCGGAACGCACCGCCATGTTAAGAGCTAAGC  
>Apoptosis+Cancer+Other_Cancer=GTF2H5_--_158589482.24-all~e39m1  
TGGGGTGTGGAACACTGAGGTTAAGAGCTAAG  
>Trafficking+Mitochondria+Motility=RHOT1_+_30469525.28-all~e39m1  
CCAGCTCCGCCTCCTCCGGCCCAGGTTAAGAGC  
>Drug_Targets+Kinase_Phosphatase=NDUFC2_+_77791131.24-all~e39m1  
GCAATTCTGGTTGGGAAGTGTAAAGAGCTAAG  
>Apoptosis+Cancer+Other_Cancer=SREBF1_+_17740225.24-all~e39m1  
GACACGAACGCGCGGAGCGGGTTAAGAGCTAAG
```

sgRNA ID
expected sequence

Running fastqgz_to_counts

run fastqgz_to_counts.py -h

```
[In [5]: run fastqgz_to_counts.py -h
usage: fastqgz_to_counts.py [-h] [-p PROCESSORS] [--trim_start TRIM_START]
                             [--trim_end TRIM_END] [--test]
                             Library_Fasta Out_File_Path Seq_File_Names
                             [Seq_File_Names ...]

Process raw sequencing data from screens to counts files in parallel

positional arguments:
  Library_Fasta      Fasta file of expected library reads.
  Out_File_Path      Directory where output files should be written.
  Seq_File_Names     Name(s) of sequencing file(s). Unix wildcards can be
                     used to select multiple files at once. The script will
                     search for all *.fastq.gz, *.fastq, and
                     *.fa(/fasta/fna) files with the given wildcard name.

optional arguments:
  -h, --help           show this help message and exit
  -p PROCESSORS, --processors PROCESSORS
  --trim_start TRIM_START
  --trim_end TRIM_END
  --test               Run the entire script on only the first 10000 reads of
                      each file. Be sure to delete or move all test files
                      before re-running script as they will not be
                      overwritten.
```

run fastqgz_to_counts.py -p 6

--trim_start 1 --trim_end 35

library_reference/CRISPRi_v1_human.trim_1_35.fa

Demo/Step1/output

Demo/Step1/Sequencing_files/Demo*

Running fastqgz_to_counts

```
run fastqgz_to_counts.py -p 6 --trim_start 1 --trim_end 35 library_reference/  
CRISPRi_v1_human.trim_1_35.fa Demo/Step1/output Demo/Step1/Sequencing_files/Demo*
```

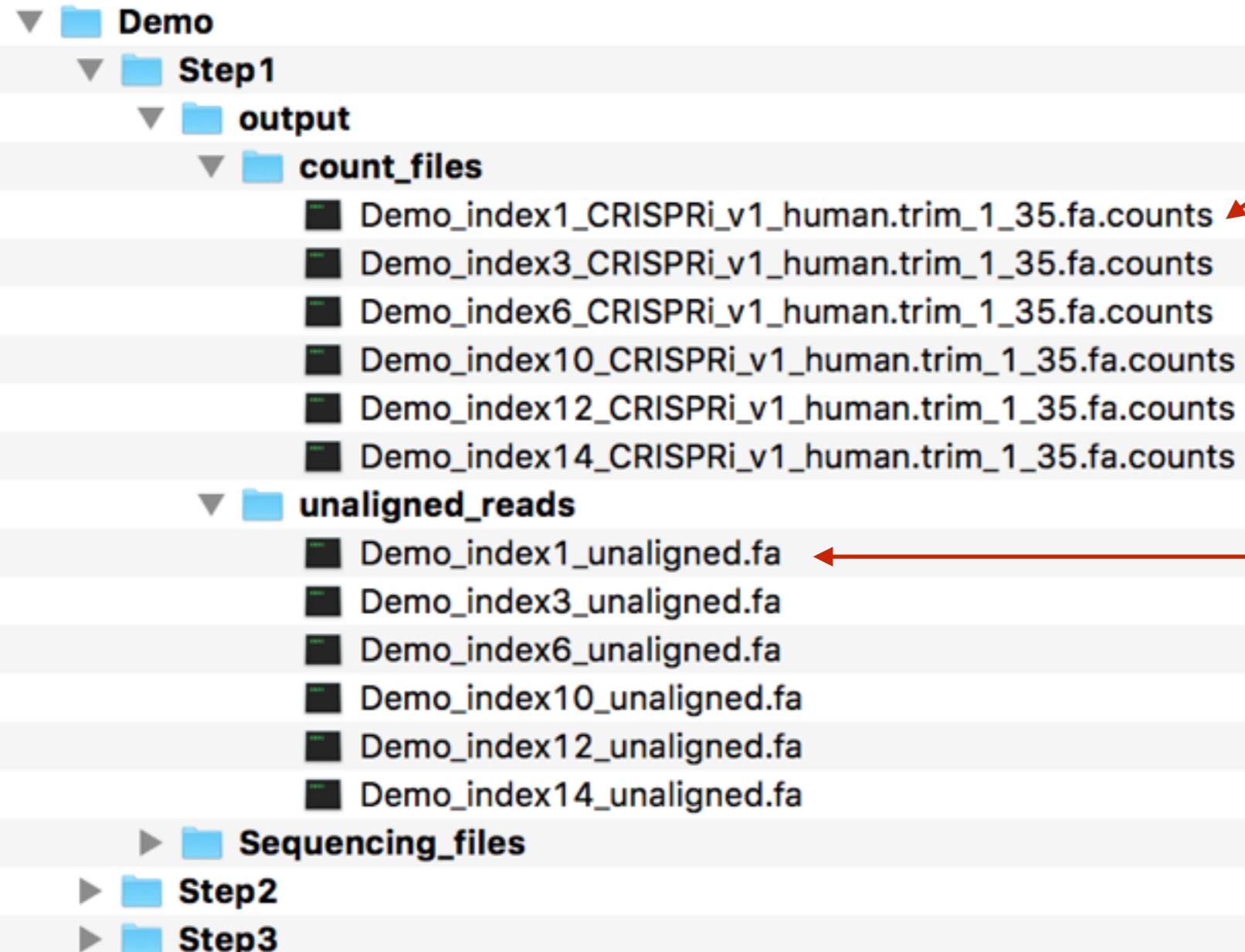
```
In [7]: run fastqgz_to_counts.py -p 6 --trim_start 1 --trim_end 35 library_reference/CRISPRi_v1_human.trim_1_35.fa Demo/Step1/output  
Demo/Step1/Sequencing_files/Demo*
```

Running fastqgz_to_counts

```
run fastqgz_to_counts.py -p 6 --trim_start 1 --trim_end 35 library_reference/  
CRISPRi_v1_human.trim_1_35.fa Demo/Step1/output Demo/Step1/Sequencing_files/Demo*
```

```
In [7]: run fastqgz_to_counts.py -p 6 --trim_start 1 --trim_end 35 library_reference/CRISPRi_v1_human.trim_1_35.fa Demo/Step1/output  
Demo/Step1/Sequencing_files/Demo*  
[Library file loaded successfully:  
    2.09E+05 elements (2.05E+05 unique sequences) 34bp reads expected  
Processing Demo/Step1/Sequencing_files/Demo_index1.fastq  
Processing Demo/Step1/Sequencing_files/Demo_index10.fastq  
Processing Demo/Step1/Sequencing_files/Demo_index12.fastq  
Processing Demo/Step1/Sequencing_files/Demo_index14.fastq  
Processing Demo/Step1/Sequencing_files/Demo_index3.fastq  
Processing Demo/Step1/Sequencing_files/Demo_index6.fastq  
Done processing Demo/Step1/Sequencing_files/Demo_index1.fastq  
Done processing Demo/Step1/Sequencing_files/Demo_index10.fastq  
Done processing Demo/Step1/Sequencing_files/Demo_index6.fastq  
Done processing Demo/Step1/Sequencing_files/Demo_index14.fastq  
Done processing Demo/Step1/Sequencing_files/Demo_index3.fastq  
Done processing Demo/Step1/Sequencing_files/Demo_index12.fastq  
Demo/Step1/output/count_files/Demo_index1_CRISPRi_v1_human.trim_1_35.fa.counts:  
    2.50E+04 reads 2.05E+04 aligning (81.99%)  
Demo/Step1/output/count_files/Demo_index10_CRISPRi_v1_human.trim_1_35.fa.counts:  
    2.50E+04 reads 2.04E+04 aligning (81.49%)  
Demo/Step1/output/count_files/Demo_index12_CRISPRi_v1_human.trim_1_35.fa.counts:  
    2.50E+04 reads 2.03E+04 aligning (81.01%)  
Demo/Step1/output/count_files/Demo_index14_CRISPRi_v1_human.trim_1_35.fa.counts:  
    2.50E+04 reads 2.03E+04 aligning (81.39%)  
Demo/Step1/output/count_files/Demo_index3_CRISPRi_v1_human.trim_1_35.fa.counts:  
    2.50E+04 reads 2.03E+04 aligning (81.31%)  
Demo/Step1/output/count_files/Demo_index6_CRISPRi_v1_human.trim_1_35.fa.counts:  
    2.50E+04 reads 2.03E+04 aligning (81.10%)  
Done processing all sequencing files
```

Running fastqgz_to_counts



```
Apoptosis+Cancer+Other_Cancer=A2M_+_9268488.25-all~e39m1
0
Apoptosis+Cancer+Other_Cancer=A2M_+_9268495.24-all~e39m1
1
Apoptosis+Cancer+Other_Cancer=A2M_+_9268513.26-all~e39m1
0
Apoptosis+Cancer+Other_Cancer=A2M_+_9268524.25-all~e39m1
0
Apoptosis+Cancer+Other_Cancer=A2M_+_9268659.23-all~e39m1
0
Apoptosis+Cancer+Other_Cancer=A2M_+_9268708.26-all~e39m1
0
Apoptosis+Cancer+Other_Cancer=A2M_+_9268728.25-all~e39m1
1
Apoptosis+Cancer+Other_Cancer=A2M_-_9268620.25-all~e39m1
0
Apoptosis+Cancer+Other_Cancer=A2M_-_9268625.25-all~e39m1
0
Apoptosis+Cancer+Other_Cancer=A2M_-_9268688.24-all~e39m1
0

>41
ACGTGGAACTGCTTACAGCGTTAACGAGCTAAGC
>77
ATGAGACGGTCTTGACTCGTTAACGAGCTAAGCT
>85
TCTTGGGCATCAGAGGTGAGAGTTAAAAGCTAA
>109
ACCAGGATGGGCACCACCCGTTAACGAGCTAAGC
>117
AACTGCCACGTCTGGTGTAAAGAGCTAAGCT
>137
AGCGCCACAACGCTCGCGTTAACGAGCTAAGCT
>169
AGGACGCAACCAAGCGGTGGTTAACGAGCTAAGC
>173
GATGAGTGAGAGAAAGAAAGTTAACGAGCTAAGC
>185
CCATGGCTTCGGGAGGCTTAGTTAAAAGCTAAG
```

Step 1: fastqgz_to_counts.py

counting sgRNAs in raw sequencing files



Step 2: process_experiments.py

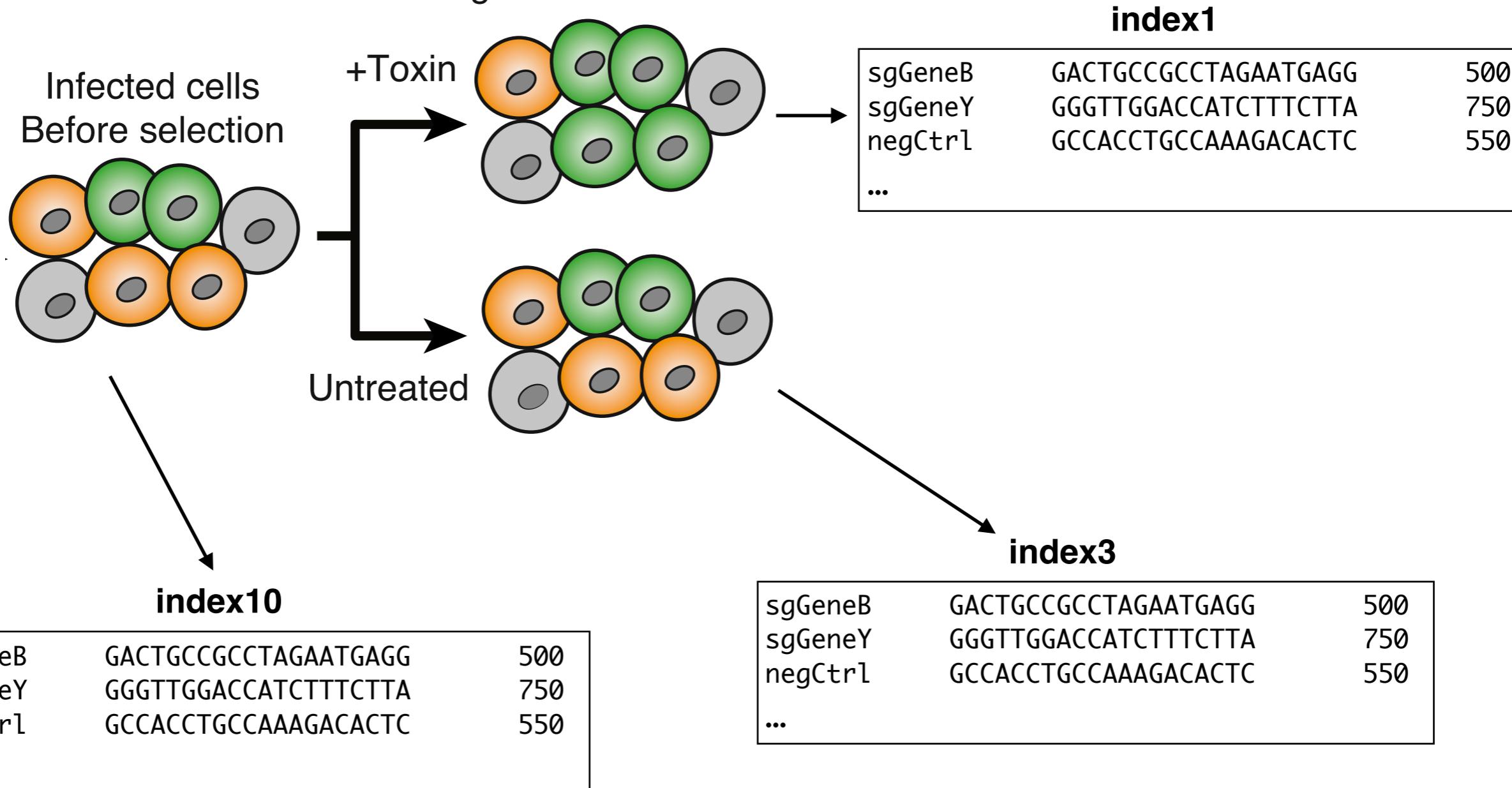
calculating sgRNA-level and gene-level phenotypes and p-values



Step 3: screen_analysis.py

making custom graphs

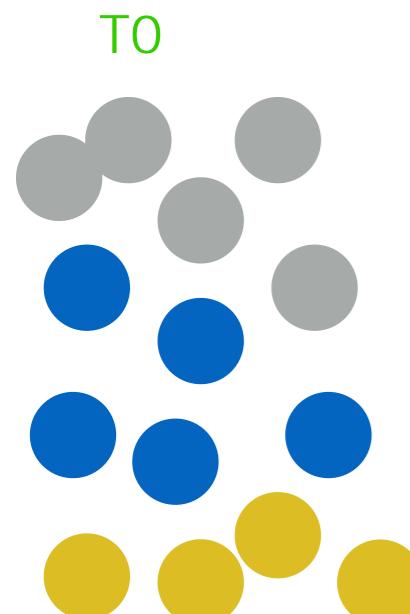
Growth analysis by
deep sequencing using
sgRNAs as barcodes



We need to compare the sgRNA read counts from different conditions to obtain quantitative phenotypes

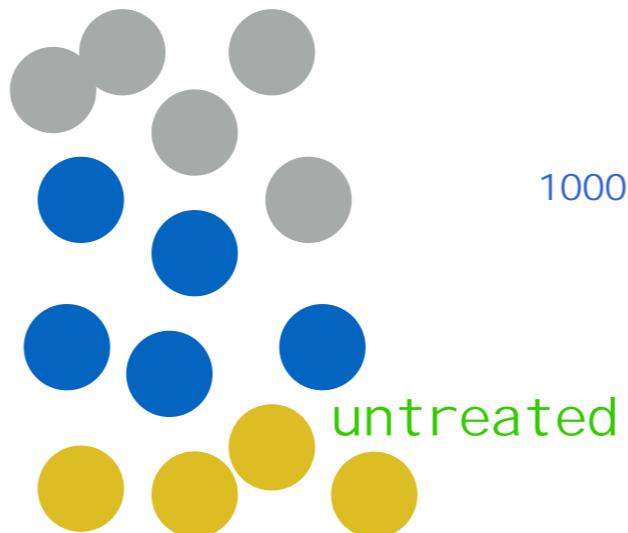
cells containing:
non-targeting sgRNA
sgRNA that promotes
resistance

sensitivity



gamma
vehicle/
no drug

+drug
tau



ρ 用来衡量药物对细胞增殖次数的影响

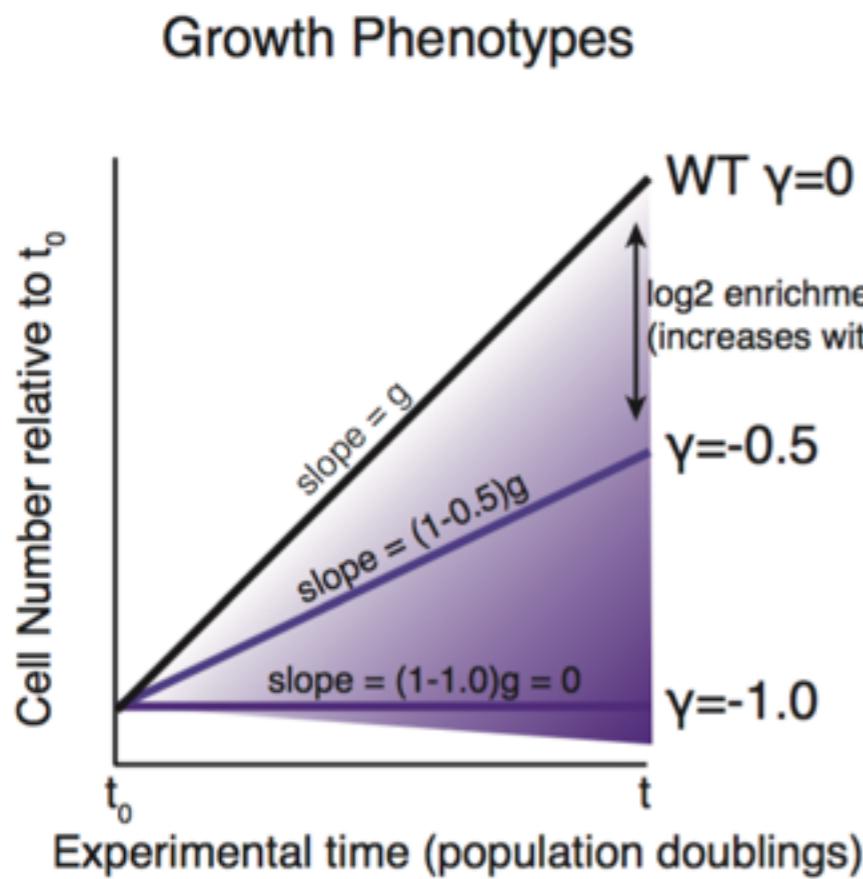
$$\rho = \log_2 \left(\frac{\text{fraction sgRNA counts, treated}}{\text{fraction sgRNA counts, untreated}} \right)$$



```
growth_value_string =  
gamma: Rep2: 11.1519761622  
rho: Rep2: 8.44158496881  
tau: Rep2: 2.7103911934  
gamma: Rep1: 10.7412001484  
rho: Rep1: 7.82935376641  
tau: Rep1: 2.9118463821
```

测序时，只取一部分细胞测序，因此无法用sgRNA的数量来推测细胞总数；
growth value使用三个细胞总数来计算，每两个样本之间都会有一个不同的growth value.

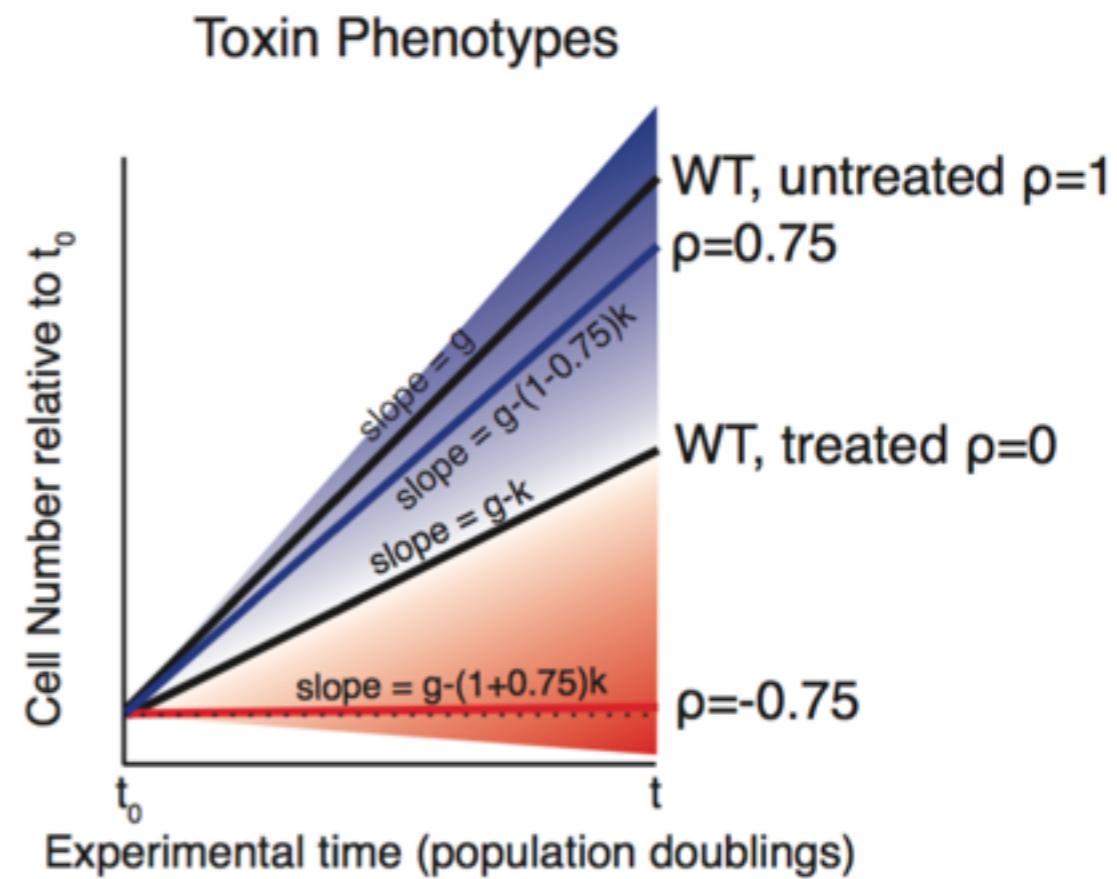
Normalizing phenotype by cell doublings



$$\text{log}_2 \text{enrichment} = \log_2(\text{Cell Fraction}_{\text{treated}} / \text{Cell Fraction}_{t_0})$$
$$\text{Normalized log}_2 e = \text{Sample log}_2 e - \text{Median NC log}_2 e$$

$$\gamma = \text{Normalized log}_2 e / t$$

Normalizer Log2e/(gt)



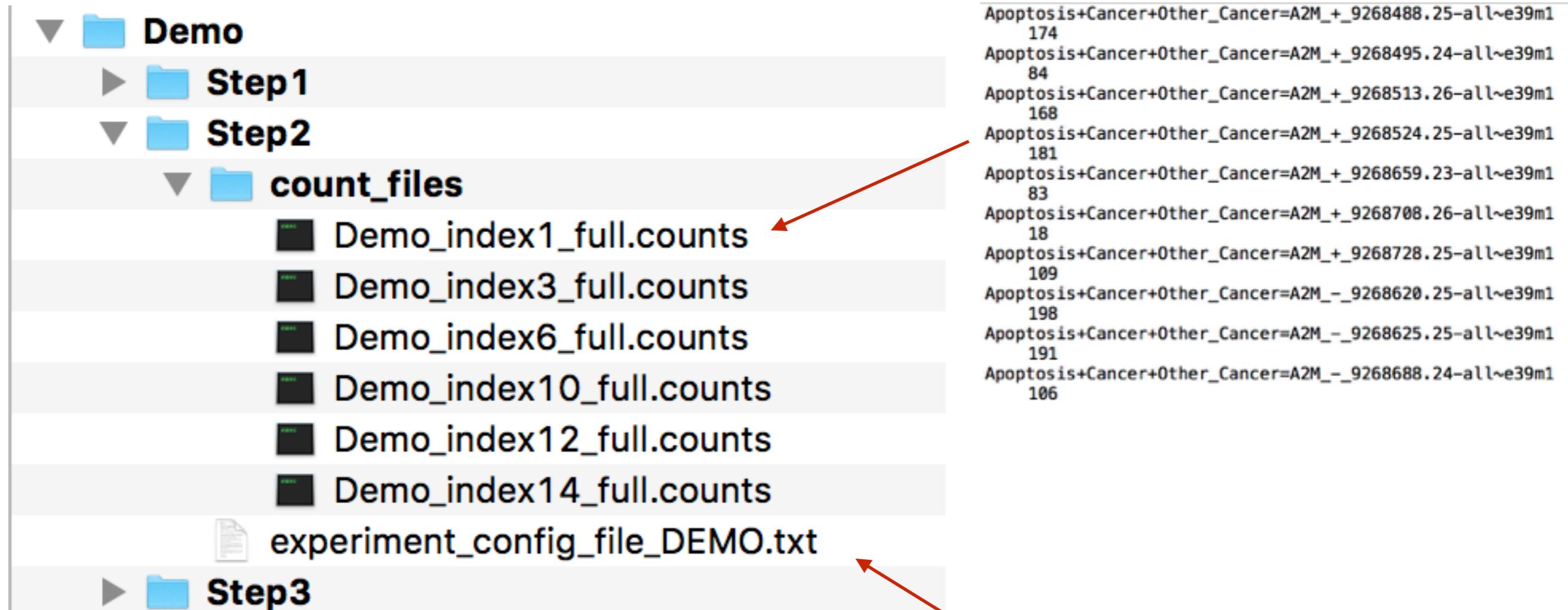
$$\text{log}_2 \text{enrichment} = \log_2(\text{Cell Fraction}_{\text{treated}} / \text{Cell Fraction}_{\text{untreated}})$$
$$\text{Normalized log}_2 e = \text{Sample log}_2 e - \text{Median NC log}_2 e$$

$$\rho = \text{Normalized log}_2 e / t$$

Normalizer Log2e/(gt)

See the templates in `cell_doubling_measurements.xlsx` to calculate t (number of cell doublings)

Setting up for process_experiments



Open the experiment_config_file
in a text editor

Setting up for process_experiments

```
#####
##          Experiment Settings      ##
#####
[experiment_settings]

output_folder = [path to Step2-counts_to_gene_scores]
experiment_name = ctx_demo

#####
##          Experiment Settings      ##
#####
[experiment_settings]

output_folder = /home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output
experiment_name = ctx_demo
```

Setting up for process_experiments

```
#####
##          Library Settings          ##
#####
[library_settings]

#currently CRISPRi_v1 or CRISPRa_v1

library = CRISPRi_v1

#If you are using all sublibraries you don't need to change anything
#Otherwise, uncomment and edit a list from below:

#CRISPRi_v1 sublibraries
sublibraries =
Apoptosis+Cancer+Other_Cancer
# Drug_Targets+Kinase_Phosphatase
# Gene_Expression
# Membrane_Proteins
# Stress_Proteostasis
# Trafficking+Mitochondria+Motility
# Unassigned
# Essential_CRISPRI

#CRISPRa_v1 sublibraries
#sublibraries =
# Apoptosis+Cancer+Other_Cancer
# Drug_Targets+Kinase_Phosphatase
# Gene_Expression
# Membrane_Proteins
# Stress_Proteostasis
# Trafficking+Mitochondria+Motility
# Unassigned
```

= ignore this line, so only the
Apoptosis+Cancer+Other_Cancer sub-library
will be analyzed to make this demo faster

Setting up for process_experiments

```
#####
##          Counts Files      ##
#####[counts_files]

#Enter the paths to counts files from fastqgz_to_counts.py
#followed by :condition|replicate_id
#For example:
#counts_file_string =
#  /home/max/counts/Sample1_index6_CRISPRi.counts:treated|Rep1
#  /home/max/counts/Sample1_index12_CRISPRi.counts:untreated|Rep1

counts_file_string =
  [index 10]:T0|Rep1
  [index 3]:untreated|Rep1
  [index 1]:treated|Rep1
  [index 12]:T0|Rep2
  [index 6]:untreated|Rep2
  [index 14]:treated|Rep2
```

这里需要改成各个文件对应的绝对路径

```
counts_file_string =
  /home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/count_files/Demo_index10_full.counts:T0|Rep1
  /home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/count_files/Demo_index3_full.counts:untreated|Rep1
  /home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/count_files/Demo_index1_full.counts:treated|Rep1
  /home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/count_files/Demo_index12_full.counts:T0|Rep2
  /home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/count_files/Demo_index6_full.counts:untreated|Rep2
  /home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/count_files/Demo_index14_full.counts:treated|Rep2

cp Demo/Step2/experiment_config_file_DEMO.txt Demo/Step2/experiment_config_fileout.txt
vim Demo/Step2/experiment_config_fileout.txt
# /home/xiongxin/github/ScreenProcessing/Demo/Step2/count_files/Demo_index10_full.counts
```

Once you've made these changes, save the file (or save as experiment_config_filledout.txt)

Setting up for process_experiments

```
#####
##      Growth Values (phenotype scores only)      ##
#####
[growth_values]
#Enter the growth values (population doublings/doubling differences)
#These values are used to normalize log2enrichments;
#default to 1 (un-normalized) if log2e is desired

#comparison_name:replicate_id:value
#For example:
#growth_value_string =
#  rho:Rep1:7.13

growth_value_string =
  gamma:Rep2:11.1519761622
  rho:Rep2:8.44158496881
  tau:Rep2:2.7103911934
  gamma:Rep1:10.7412001484
  rho:Rep1:7.82935376641
  tau:Rep1:2.9118463821
```

Note that this demo has pre-filled values for the growth_value_string

- This allows you to normalize phenotypes by cell doublings (see previous slides)
- Growth values can be calculated using the cell doubling 这个需要检查一下 measurements spreadsheet
- This normalization is optional for growth screens and may not be applicable for other types of screens

Once you've made these changes, save the file (or save as experiment_config_filledout.txt)

Running process_experiments

```
[In [8]: run process_experiments.py -h
```

```
usage: process_experiments.py [-h] Config_File Library_File_Directory

Calculate sgRNA- and gene-level phenotypes based on sequencing read counts, as
specified by the experiment config file.

positional arguments:
  Config_File      Experiment config file specifying screen analysis
                    settings (see accompanying BLANK and DEMO files).
  Library_File_Directory
                    Directory containing reference library tables and the
                    library_config.txt file.

optional arguments:
  -h, --help        show this help message and exit
```

Running process_experiments

```
[In [8]: run process_experiments.py -h
```

```
usage: process_experiments.py [-h] Config_File Library_File_Directory

Calculate sgRNA- and gene-level phenotypes based on sequencing read counts, as
specified by the experiment config file.

positional arguments:
  Config_File      Experiment config file specifying screen analysis
                    settings (see accompanying BLANK and DEMO files).
  Library_File_Directory
                    Directory containing reference library tables and the
                    library_config.txt file.

optional arguments:
  -h, --help        show this help message and exit
```

```
[In [9]: run process_experiments.py Demo/Step2/experiment_config_file_DEMO_filledout.txt library_tables/
```

Running process_experiments

```
[In [8]: run process_experiments.py -h
```

```
usage: process_experiments.py [-h] Config_File Library_File_Directory

Calculate sgRNA- and gene-level phenotypes based on sequencing read counts, as
specified by the experiment config file.

positional arguments:
  Config_File      Experiment config file specifying screen analysis
                    settings (see accompanying BLANK and DEMO files).
  Library_File_Directory
                    Directory containing reference library tables and the
                    library_config.txt file.

optional arguments:
  -h, --help        show this help message and exit
```

```
[In [9]: run process_experiments.py Demo/Step2/experiment_config_file_DEMO_filledout.txt library_tables/
Accessing library information
Loading counts data
Merging experiment counts split across lanes/indexes
```

Running process_experiments

```
[In [8]: run process_experiments.py -h
```

```
usage: process_experiments.py [-h] Config_File Library_File_Directory

Calculate sgRNA- and gene-level phenotypes based on sequencing read counts, as
specified by the experiment config file.

positional arguments:
  Config_File      Experiment config file specifying screen analysis
                    settings (see accompanying BLANK and DEMO files).
  Library_File_Directory  Directory containing reference library tables and the
                           library_config.txt file.

optional arguments:
  -h, --help        show this help message and exit
```

```
python process_experiments.py Demo/Step2/experiment_config_file_DEMO.txt library_tables/
```

```
[In [9]: run process_experiments.py Demo/Step2/experiment_config_file_DEMO_filledout.txt library_tables/
```

```
Accessing library information
Loading counts data
Merging experiment counts split across lanes/indexes
```

Once you see that the script is running,
it will take ~5 minutes to complete...

这一步运行报错了，需要debug
20190418

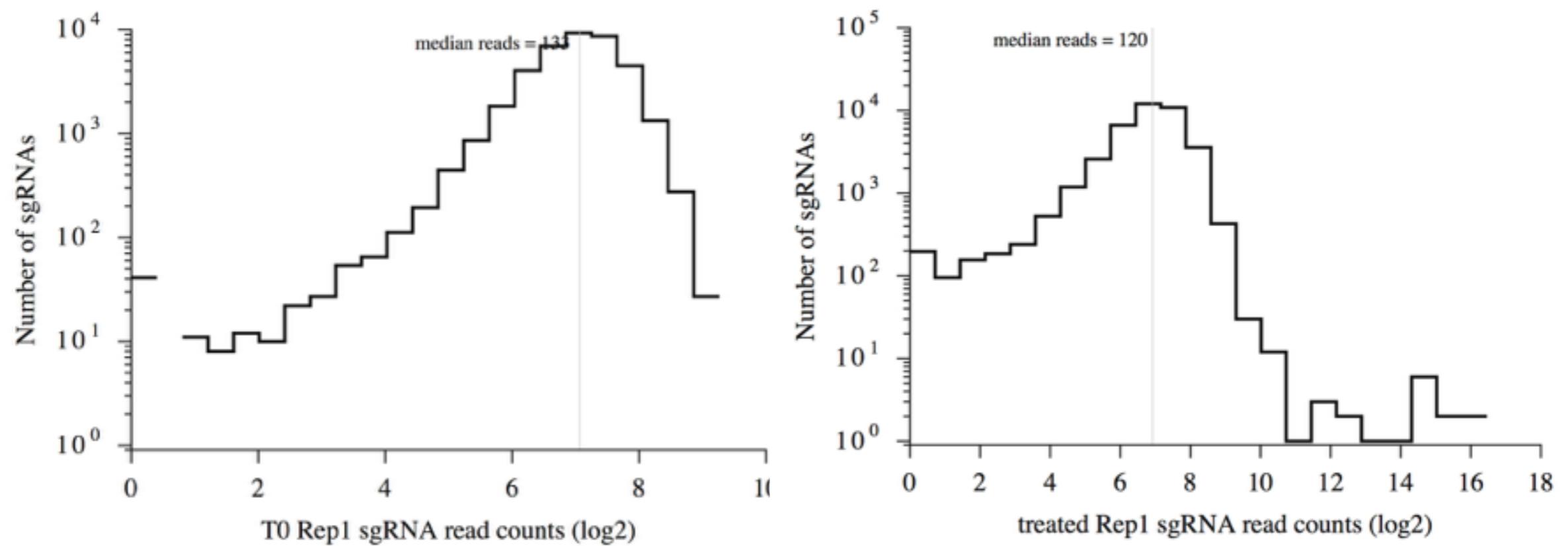
Running process_experiments

```
[In [9]: run process_experiments.py Demo/Step2/experiment_config_file_DEMO_filledout.txt library_tables/  
  
Accessing library information  
Loading counts data  
Merging experiment counts split across lanes/indexes  
-generating sgRNA read count histograms  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/000_fig_counts_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/001_fig_counts_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/002_fig_counts_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/003_fig_counts_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/004_fig_counts_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/005_fig_counts_hist.svg  
Computing sgRNA phenotype scores  
-generating phenotype histograms and scatter plots  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/006_fig_counts_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/007_fig_phenotype_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/008_fig_sgRNAs_passing_filter_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/009_fig_counts_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/010_fig_phenotype_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/011_fig_sgRNAs_passing_filter_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/012_fig_counts_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/013_fig_phenotype_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/014_fig_sgRNAs_passing_filter_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/015_fig_counts_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/016_fig_phenotype_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/017_fig_sgRNAs_passing_filter_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/018_fig_counts_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/019_fig_phenotype_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/020_fig_sgRNAs_passing_filter_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/021_fig_counts_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/022_fig_phenotype_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/023_fig_sgRNAs_passing_filter_hist.svg  
Averaging replicates  
-generating replicate phenotype histograms and scatter plots  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/024_fig_phenotype_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/025_fig_phenotype_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/026_fig_phenotype_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/027_fig_phenotype_hist.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/028_fig_phenotype_scatter.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/029_fig_phenotype_hist.svg  
Generating a pseudogene distribution from negative controls  
Computing gene scores  
--calculate_ave  
--calculate_mw  
Collapsing transcript scores to gene scores  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/030_fig_volcano_plot.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/031_fig_volcano_plot.svg  
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/032_fig_volcano_plot.svg  
Done!
```

Running process_experiments

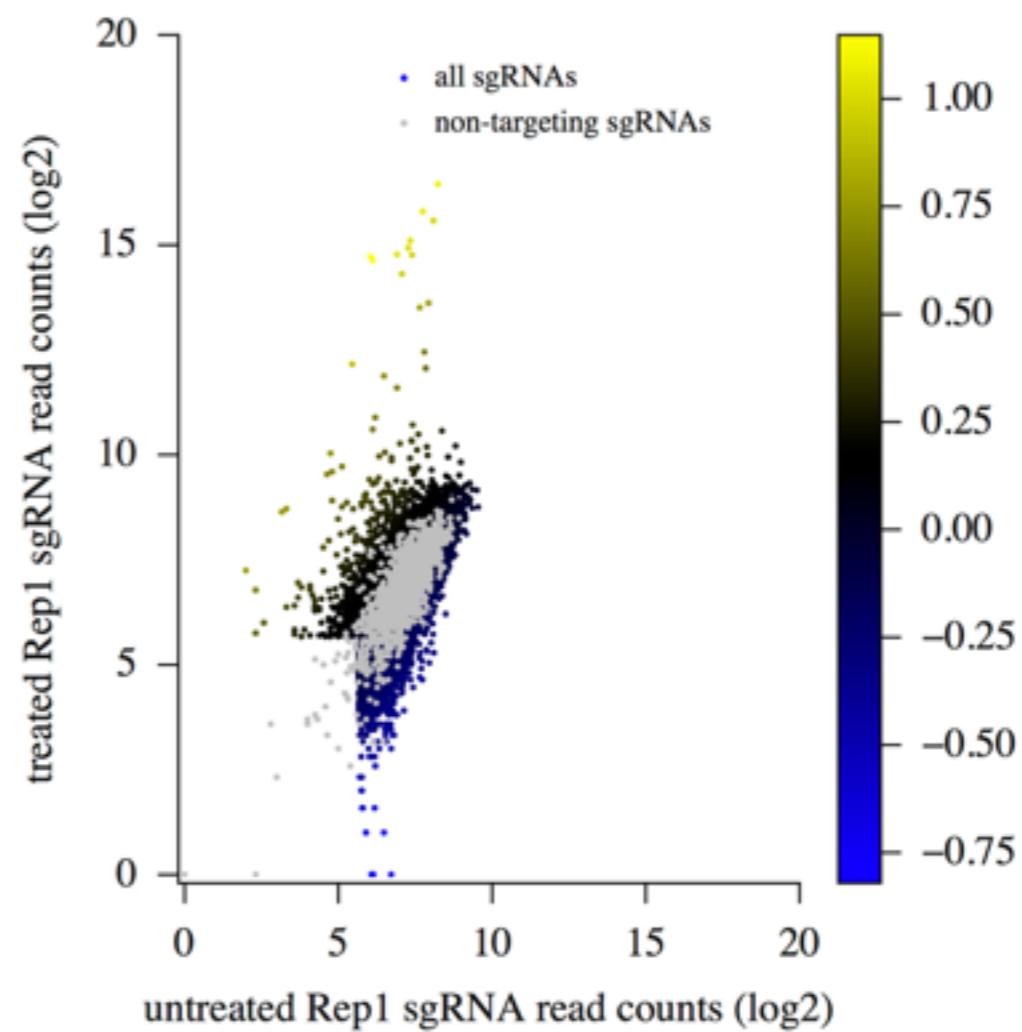
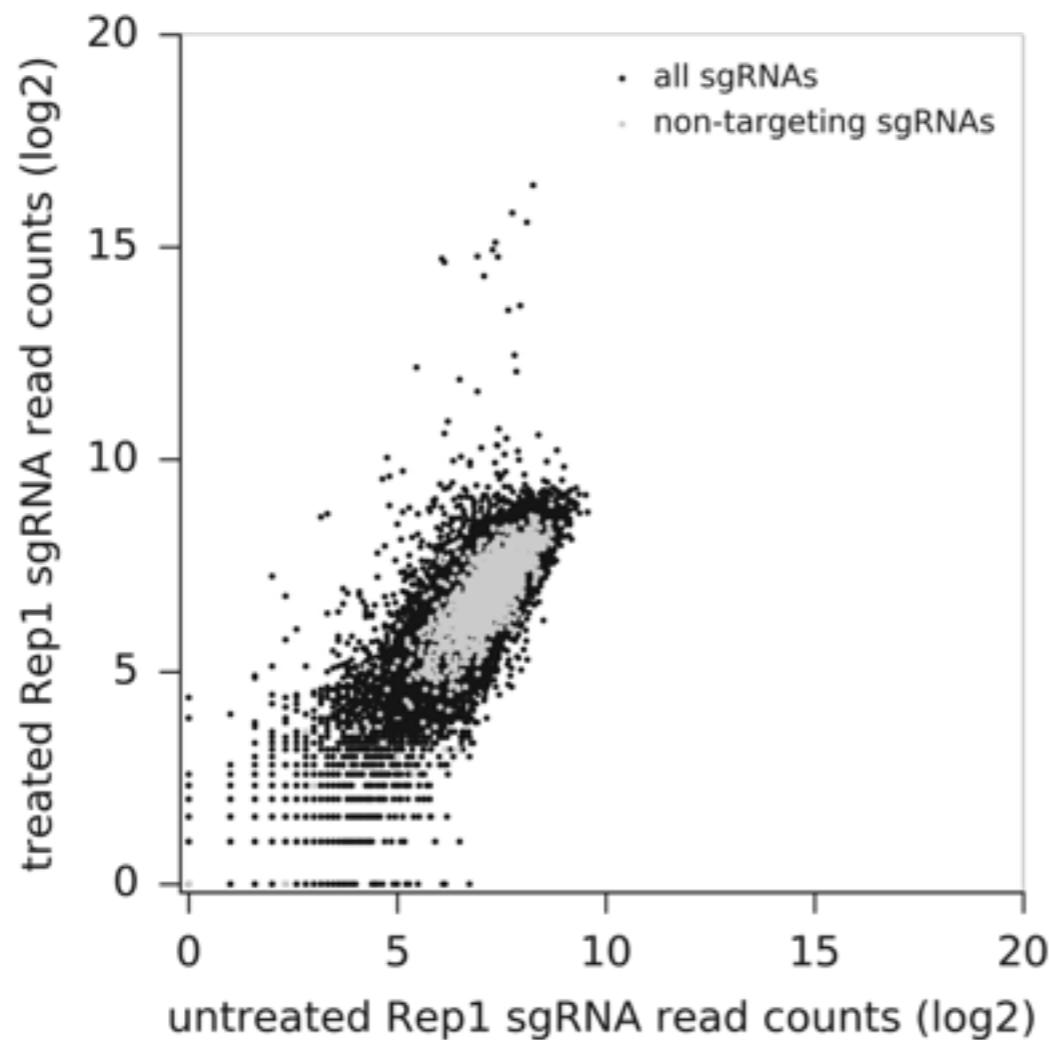
```
[In [9]: run process_experiments.py Demo/Step2/experiment_config_file_DEMO_filledout.txt library_tables/
```

```
Accessing library information
Loading counts data
Merging experiment counts split across lanes/indexes
-generating sgRNA read count histograms
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/000_fig_counts_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/001_fig_counts_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/002_fig_counts_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/003_fig_counts_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/004_fig_counts_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/005_fig_counts_hist.svg
```



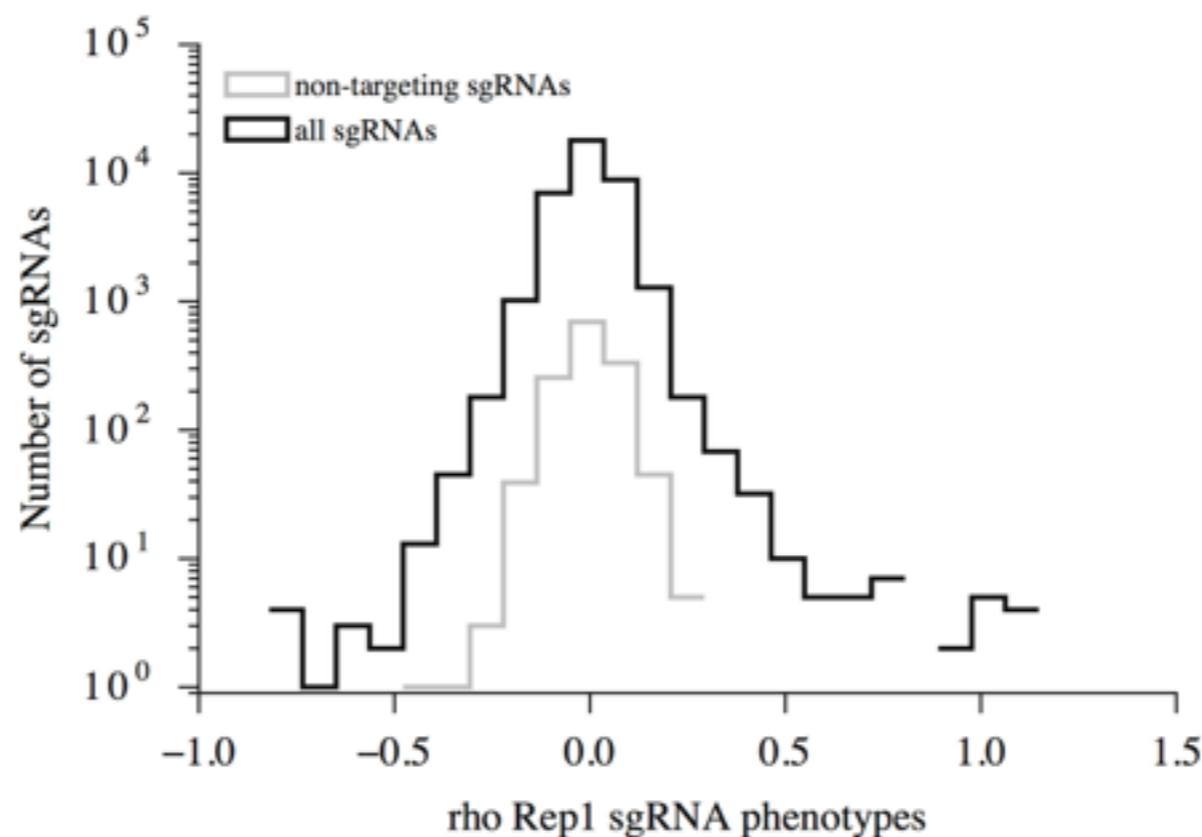
Running process_experiments

```
Computing sgRNA phenotype scores
-generating phenotype histograms and scatter plots
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/006_fig_counts_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/007_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/008_fig_sgRNAs_passing_filter_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/009_fig_counts_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/010_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/011_fig_sgRNAs_passing_filter_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/012_fig_counts_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/013_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/014_fig_sgRNAs_passing_filter_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/015_fig_counts_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/016_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/017_fig_sgRNAs_passing_filter_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/018_fig_counts_scatter.svg
```



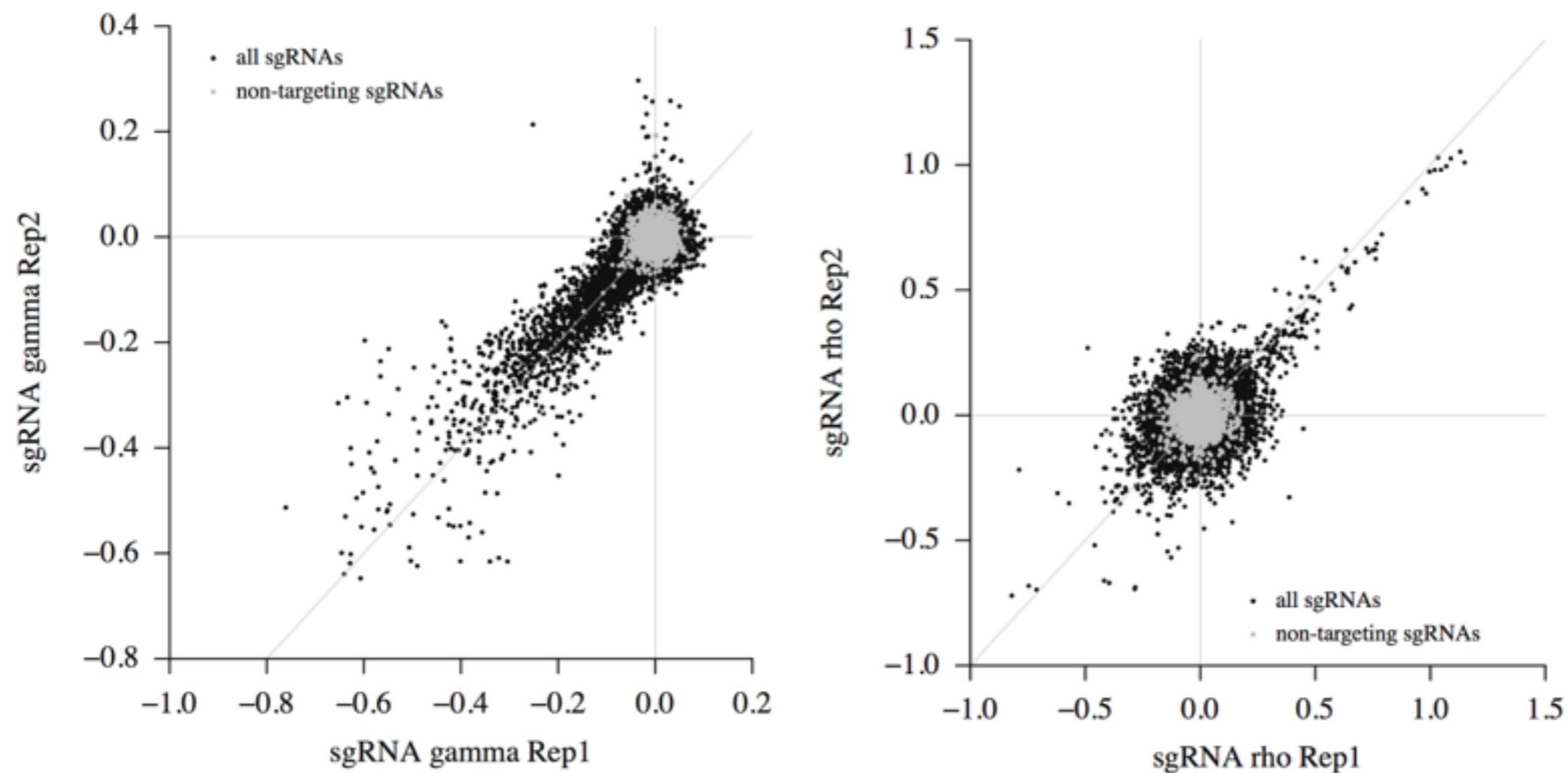
Running process_experiments

```
Computing sgRNA phenotype scores
-generating phenotype histograms and scatter plots
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/006_fig_counts_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/007_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/008_fig_sgRNAs_passing_filter_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/009_fig_counts_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/010_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/011_fig_sgRNAs_passing_filter_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/012_fig_counts_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/013_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/014_fig_sgRNAs_passing_filter_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/015_fig_counts_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/016_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/017_fig_sgRNAs_passing_filter_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/018_fig_counts_scatter.svg
```



Running process_experiments

```
Averaging replicates
-generating replicate phenotype histograms and scatter plots
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/024_fig_phenotype_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/025_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/026_fig_phenotype_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/027_fig_phenotype_hist.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/028_fig_phenotype_scatter.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/029_fig_phenotype_hist.svg
```

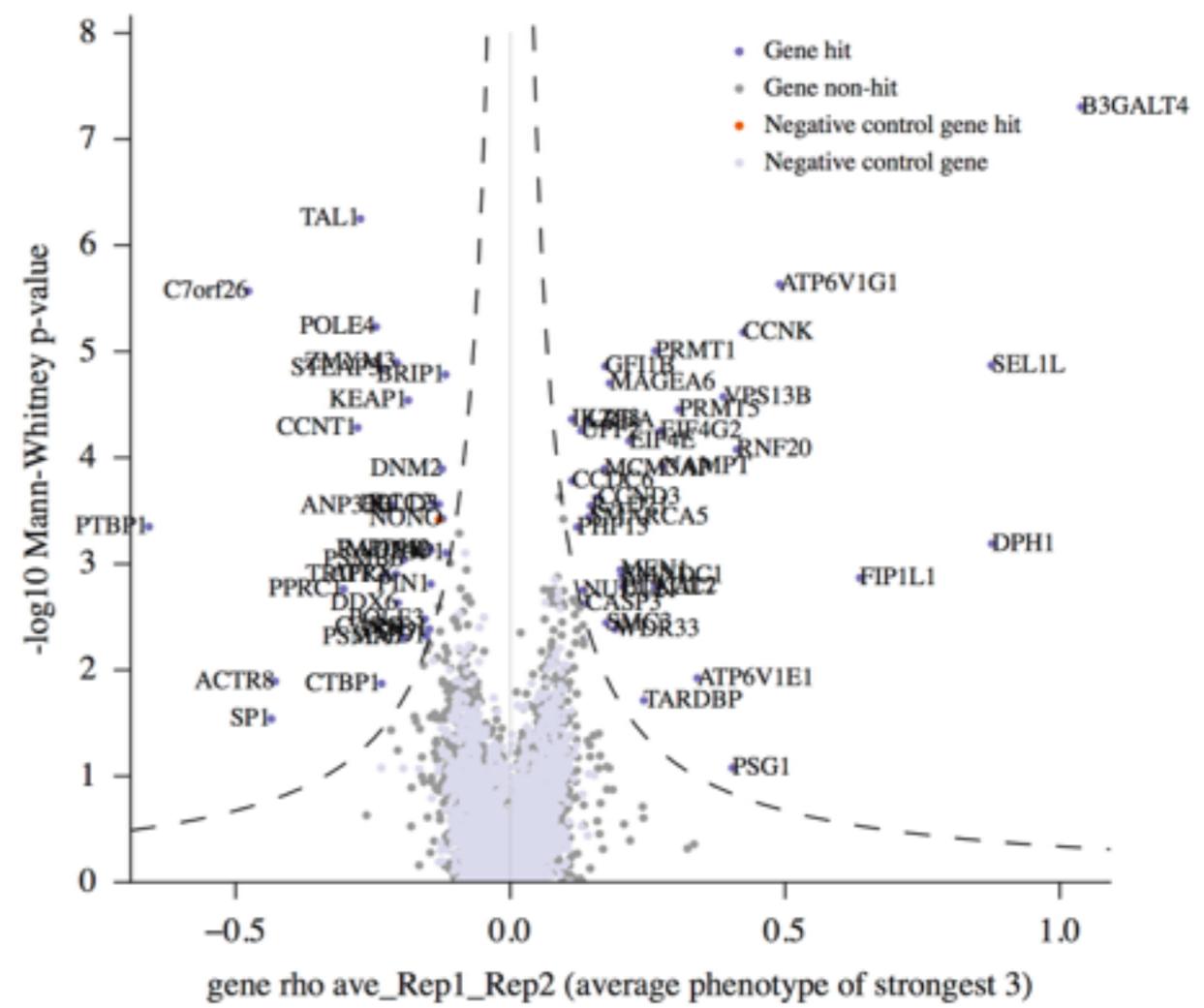
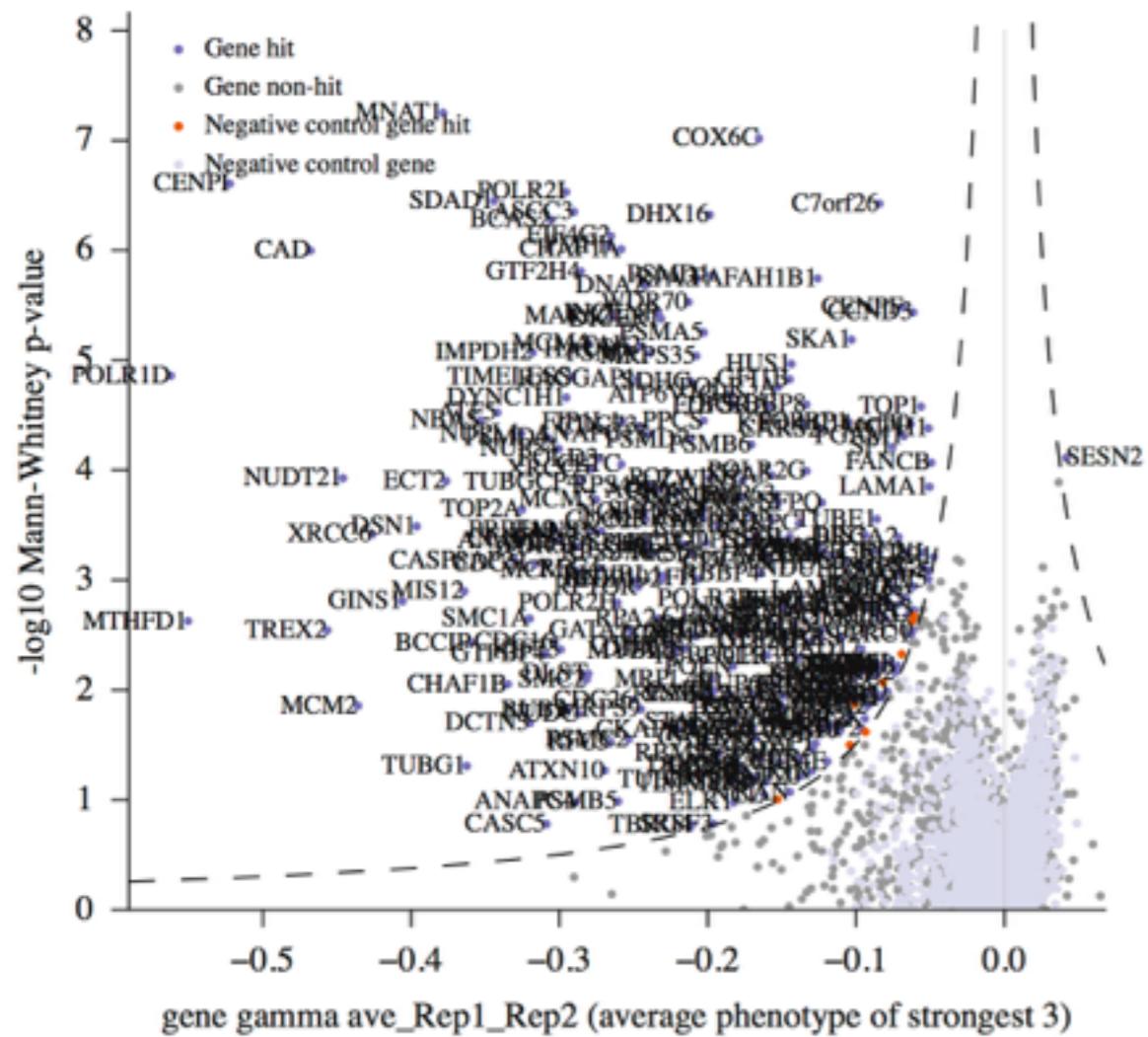


Running process_experiments

```

Generating a pseudogene distribution from negative controls
Computing gene scores
--calculate_ave
--calculate_mw
Collapsing transcript scores to gene scores
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/030_fig_volcano_plot.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/031_fig_volcano_plot.svg
/home/mhorlbeck/home_tmp/mhorlbeck/ScreenProcessing/Demo/Step2/output/ctx_demo_plots/032_fig_volcano_plot.svg
Done!

```



Step 1: fastqgz_to_counts.py

counting sgRNAs in raw sequencing files



Step 2: process_experiments.py

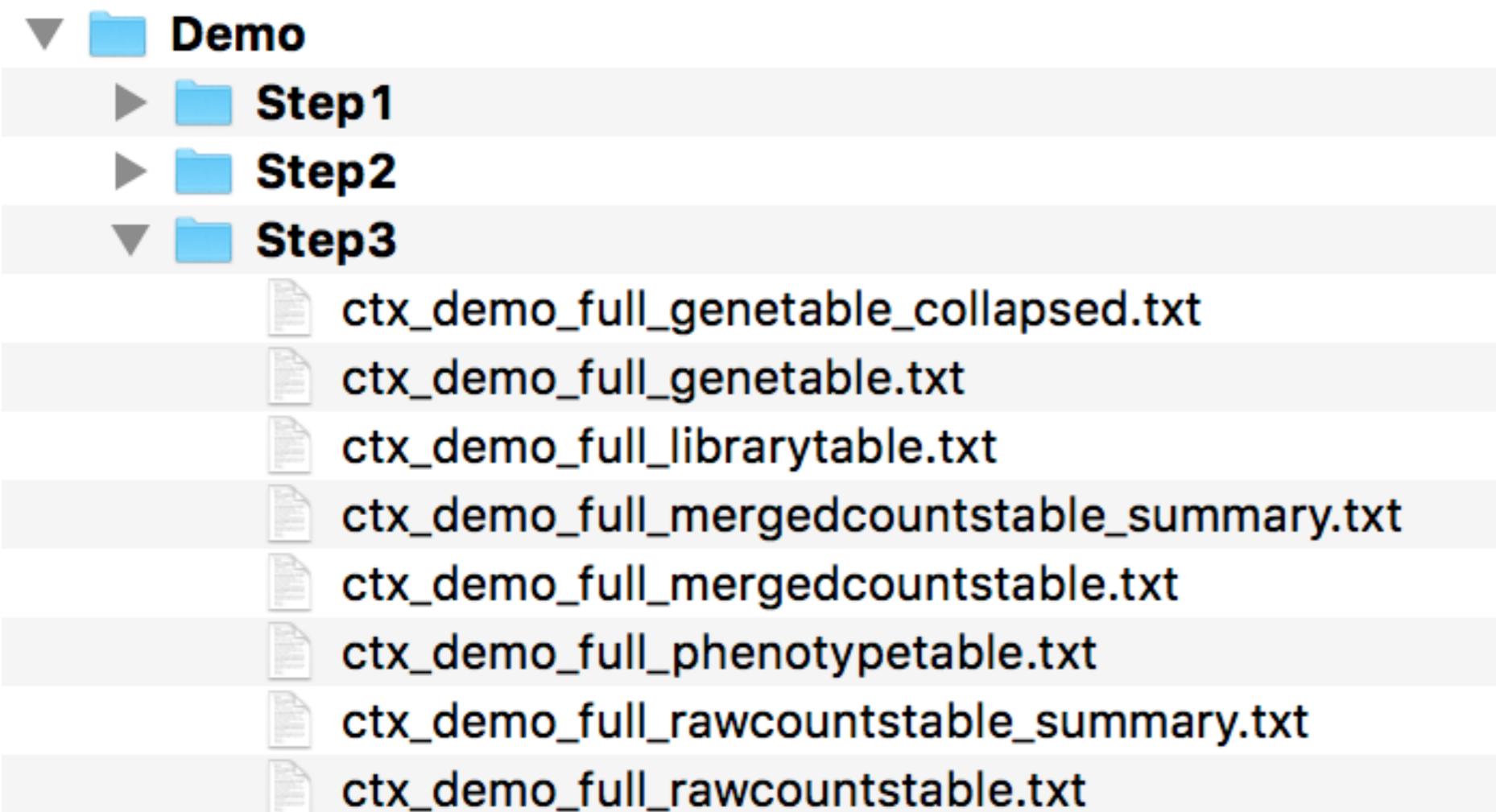
calculating sgRNA-level and gene-level phenotypes and p-values



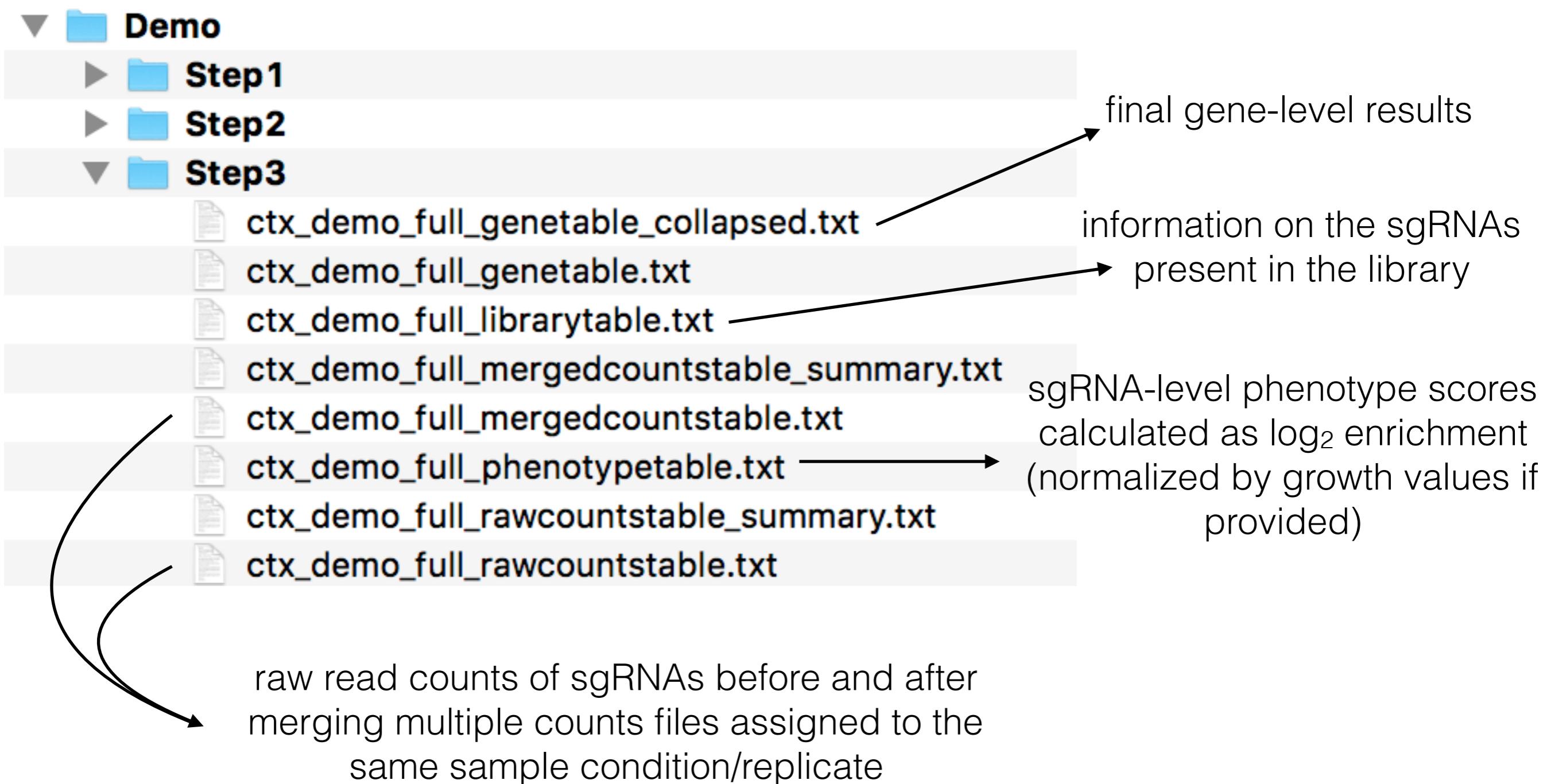
Step 3: screen_analysis.py

making custom graphs

process_experiments resulted in a lot of data tables....



...which can be explored using
excel or similar programs



Interpreting genetable_collapsed

A	B	C	D	E	F
1	gamma	gamma	gamma	gamma	gamma
2	Rep1	Rep1	Rep1	Rep1	Rep1
3	transcripts	Mann-Whitney p-value	average phenotype of strongest 3	sgRNA count_MW	sgRNA count_avg
4	gene				
5	A1BG	all	0.617489319	0.008079268	10
6	A1CF	ENST00000374001.2,EI	0.774850008	-0.002736356	9
7	A2M	all	0.072458779	0.013644911	10
8	A2ML1	all	0.739439742	-0.01119083	9
9	A4GALT	ENST00000249005.2,EI	0.499943531	-0.00265974	10

Transcription start site with lowest-scoring p-value, where multiple exist (all transcripts are listed in _genetable.txt)

The phenotypes of all sgRNAs targeting the gene are ranked by absolute value, and the top 3 are averaged to obtain a gene-level phenotype

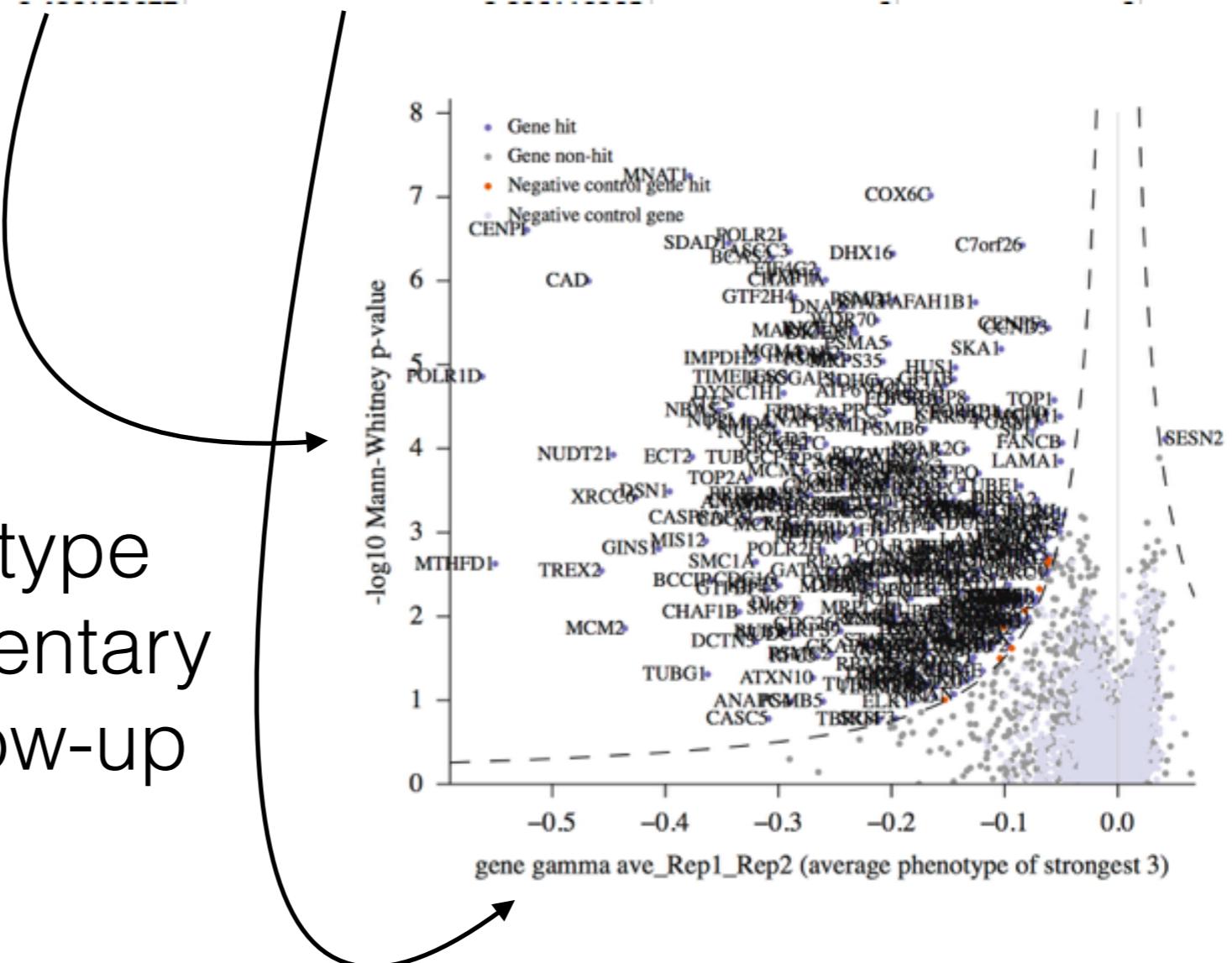
Mann-Whitney p-value comparing the phenotypes of all sgRNAs targeting the gene to all negative control sgRNAs

Number of sgRNAs included in the library and passing the read count filter

Interpreting genetable_collapsed

A	B	C	D	E	F
1	gamma	gamma	gamma	gamma	gamma
2	Rep1	Rep1	Rep1	Rep1	Rep1
3	transcripts	Mann-Whitney p-value	average phenotype of strongest 3	sgRNA count_MW	sgRNA count_avg
4	gene				t
5	A1BG	all	0.617489319	0.008079268	10
6	A1CF	ENST00000374001.2,EI	0.774850008	-0.002736356	9
7	A2M	all	0.072458779	0.013644911	10
8	A2ML1	all	0.739439742	-0.01119083	9
9	A4GALT	ENST00000249005.2,EI	0.499943531	-0.00265974	10

The p-value and phenotype scores provide complementary information to guide follow-up of screen hits



screen_analysis offers a number of functions to generate new graphs

```
[In [10]: run screen_analysis.py
```

```
[In [11]: data = loadData('Demo/Step3/ctx_demo_full')
```

all functions follow the format:
graphName(data, phenotype, replicate, ...)

and you can enter graphName(data) to find out what the available phenotype/replicate options are for the plot

```
[In [12]: countsHistogram(data)
Condition and Replicate options are:
T0           Rep1
T0           Rep2
treated      Rep1
treated      Rep2
untreated    Rep1
untreated    Rep2
```

Available plotting functions

loadData(experimentName, collapsedToTranscripts = True, premergedCounts = False)

countsHistogram(data, **condition**=None, **replicate**=None)

countsScatter(data, **condition_x** = None, **replicate_x** = None,
condition_y = None, **replicate_y** = None,
showAll = True, showNegatives = True, showGenes = [],
colorByPhenotype_condition = None, colorByPhenotype_replicate = None)

premergedCountsScatterMatrix(data, condition=None, replicate=None)

phenotypeHistogram(data, **phenotype**=None, **replicate**=None)

phenotypeScatter(data, **phenotype_x** = None, **replicate_x** = None,
phenotype_y = None, **replicate_y** = None,
showAll = True, showNegatives = True,
showGenes = [], showGeneSets = {})

sgRNAsPassingFilterHist(data, phenotype, replicate, transcripts=False)

volcanoPlot(data, **phenotype**=None, **replicate**=None, transcripts=False, showPseudo=True,
effectSizeLabel=None, pvalueLabel=None, hitThreshold=7,
labelHits = False, showGeneSets = {}, labelGeneSets = True)