

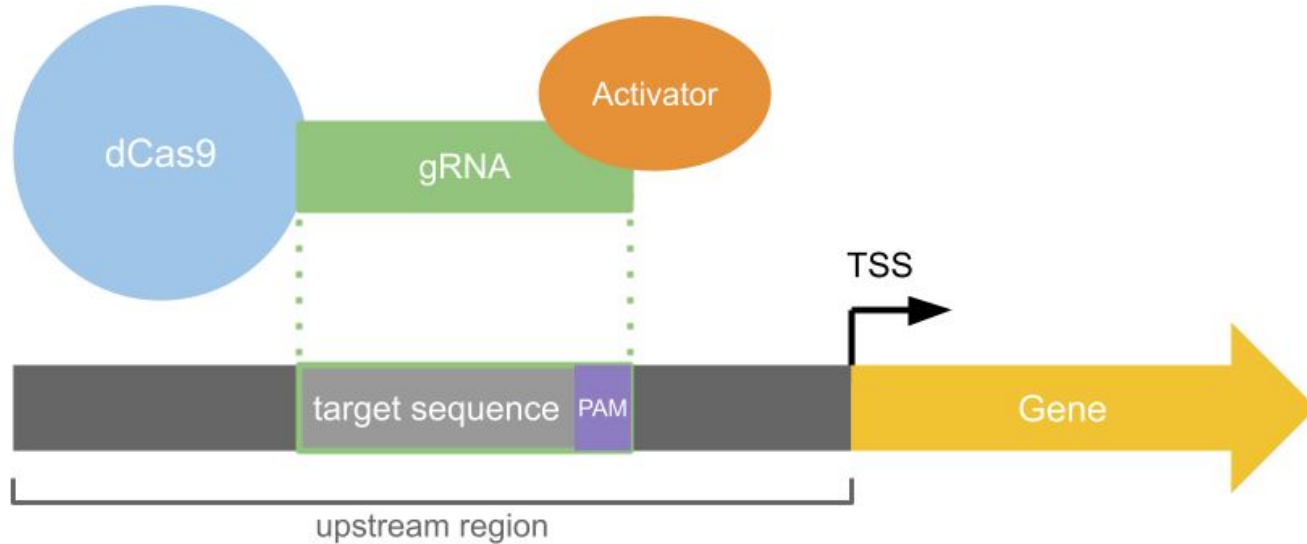
Data Analysis and Software Design to Assist Researchers in Choosing Effective Endogenous Genes for CRISPRa

Presented by Joely Nelson

Thesis Goal: To help gather and compile rules for endogenous CRISPR activation using data driven techniques

Background

What is CRISPRa?



A **dCas9**, **gRNA**, and an **activator** come together to sit on a **target sequence** upstream of a **gene** and transcription start site (**TSS**) to promote transcription. The Protospacer-Adjacent-Motif (**PAM**) is the sequence immediately next to the target sequence, usually consisted of 3 nucleotides, and is required for dCas9 recognition.

Endogenous CRISPRa

- Endogenous CRISPRa: Targeting genes with CRISPRa that already exist natively in an organism's genome.
- Using endogenous CRISPRa has the the potential to take advantage and improve already existing metabolic pathways to control metabolic production

CRISPRa's Stringent Rules....

- Changing the sequence of the gRNA can have a dramatic impact on the effectiveness of the CRISPRa
 - This means there are limits to which genes CRISPRa can effectively target

So, how can we tell if a gene is a good candidate for CRISPRa?

Using Known Rules

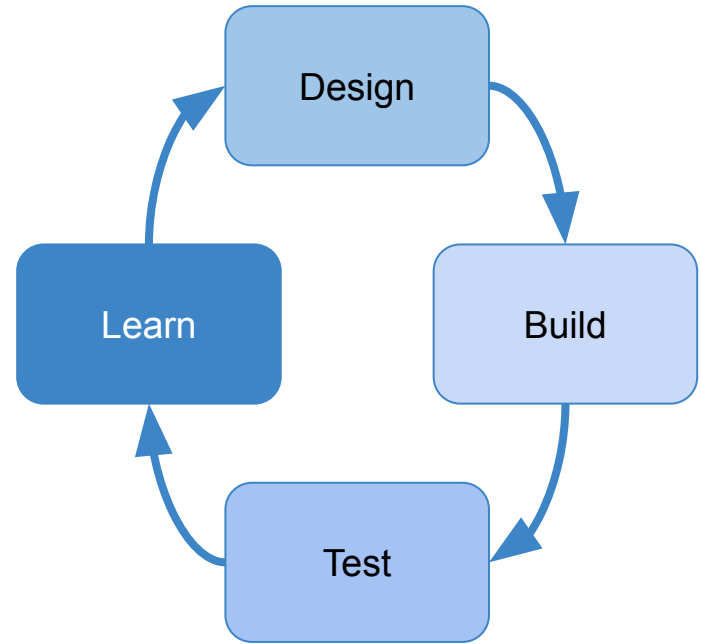
- Some known rules:
 - the baseline expression
 - the distance from the endogenous transcription start site
 - the recognition sequences (i.e. protospacer adjacent motifs, or PAMs) that can be targeted.
- These rules have not been collected and combined into a single model that can be applied to multiple genomes.
 - Some of my prior work involved incorporating some rules (distance from TSS, PAM sequence), but not all (did not include baseline expression)

Uncovering New Rules

- We don't know all the rules for CRISPRa
- We should be able to learn more through experimentation, data analysis, and perhaps even machine learning!
- But before that we need lots of data and we need to understand that data.

Data Driven Approaches to Tackle These Problems

- Analysis techniques can
 - Help us uncover information.
 - Provide us guidelines for future designs
 - Prevent us from wasting time spent with trial and error
- Design and learn Part of the Design-Build-Test-Learn cycle
- Take data from the results of experiments to learn new rules, and suggest future designs



My thesis projects

Uncovering new rules motivates my first project: **Mock Data Generation for FACS-Seq CRISPRa**

Leveraging and compiling existing rules motivates my second project: ***P. putida* Gene Expression CRISPRa Filtering**

Project 1: Mock Data Generation for FACS-Seq CRISPRa

Overview

- We had an experimental design to get expression levels for various different CRISPRa guides
- We planned to perform data analysis and machine learning techniques on this data to uncover more CRISPRa rules
- Due to COVID the experiment was delayed

Project: Create synthetic data that mimicked the results we expected to get from the experiment in order to have a starting point for developing machine learning and data analysis techniques.

The Experiment

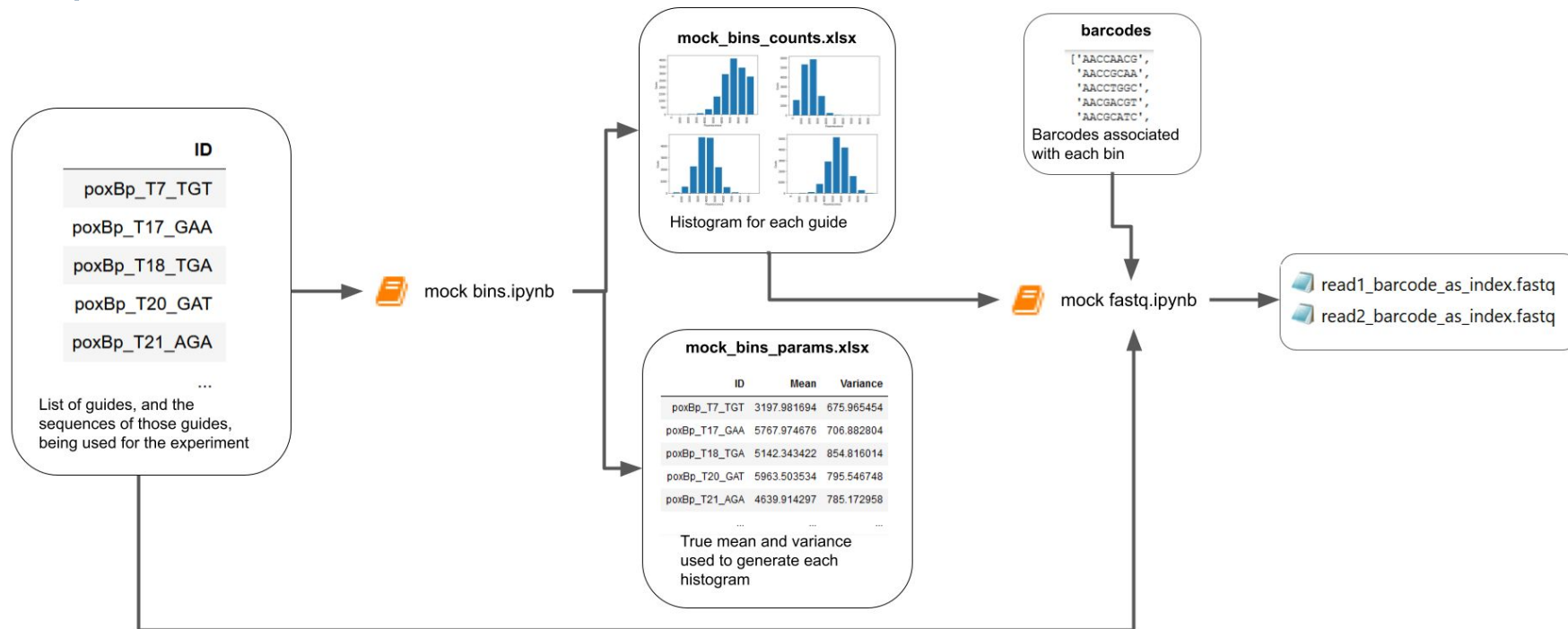
- 1,700 different guide variants, 15,000 replicates per guide
 - Many guides from the same library with a slightly different sequence
- Fluorescence taken as a measure of effectiveness
- Cells would be binned using fluorescence-activated cell sorting (FACS)
 - Cells binned based on their fluorescence level
 - Each cell in a particular bin would have a barcode added to it. Looking at this barcode will determine which bin the cell was sorted into.
- Populations would be sequenced to inform which guide was present in each bin

Ambiguous Data

- It was somewhat ambiguous what data would be produced
 - Because we had not worked with it before
- Having synthetic data available could help with the development of analysis pipelines *before* the experiment was carried out.

I designed and wrote software that would generate this synthetic data

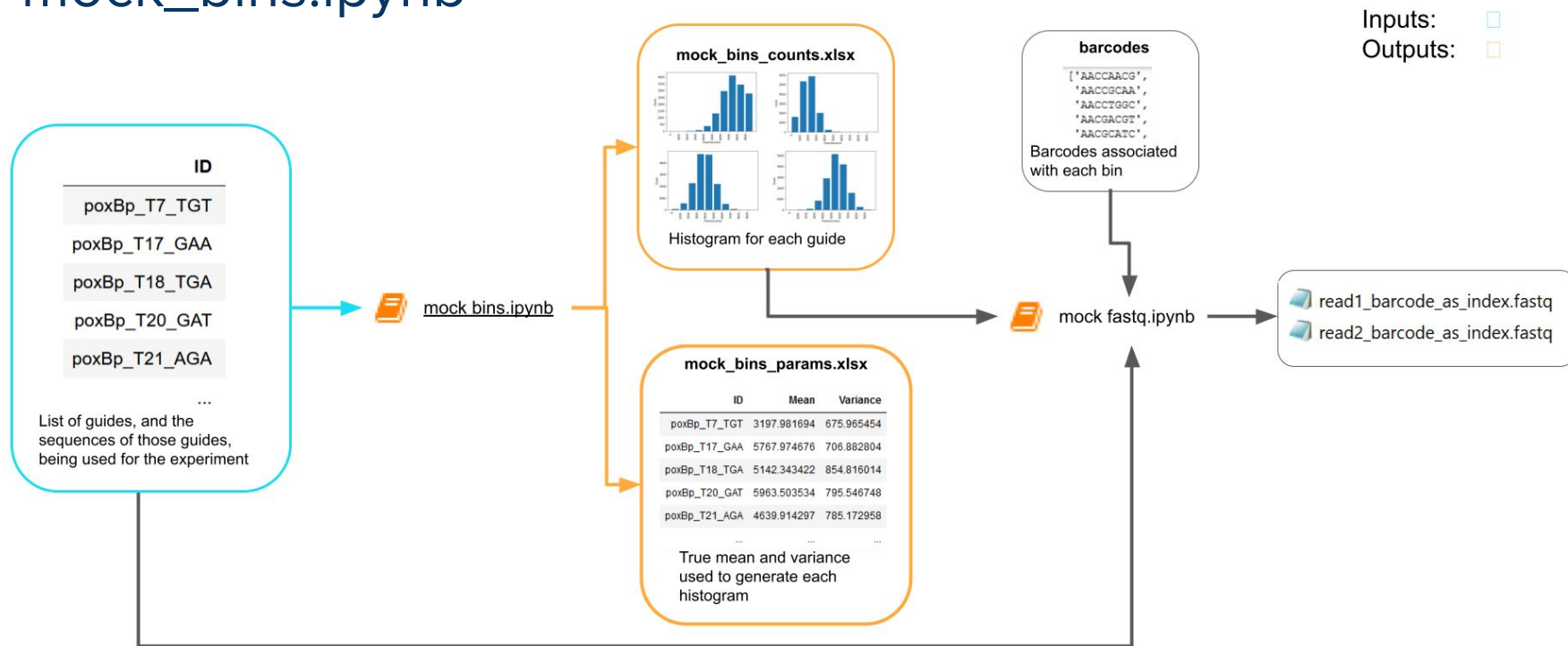
Pipeline



Mock Bins Generation

- The first step is a program called `mock_bins.ipynb` which will generate a binned histogram of the fluorescence expected to be output by the FACS process for the various guides used in the experiment
- The data created from this program would not be representative of any real data file that would be available for researchers

mock_bins.ipynb



Fluorescent Distribution Model

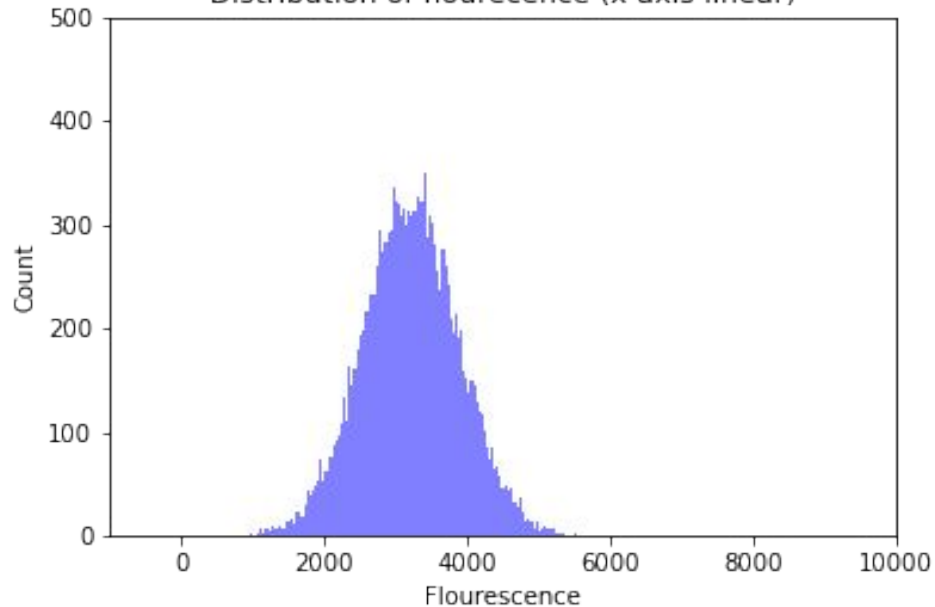
- For each of the 1700 guide variants i , It's fluorescence is modeled to come from a Gaussian distribution, F_i
 - $F_i - N(\mu_i, \sigma_i^2)$
- Each μ_i and σ_i were generated from a uniform distribution
 - $\mu_i - U(2750, 7000)$
 - $\sigma_i - U(600, 900)$
 - The Uniform distributions can be specified by the user, but are defaulted to these values to create distributions similar in shape to real data

- So, for each cell, let its guide variant be i , it's fluorescence will be a sample drawn from $N(\mu_i, \sigma_i^2)$

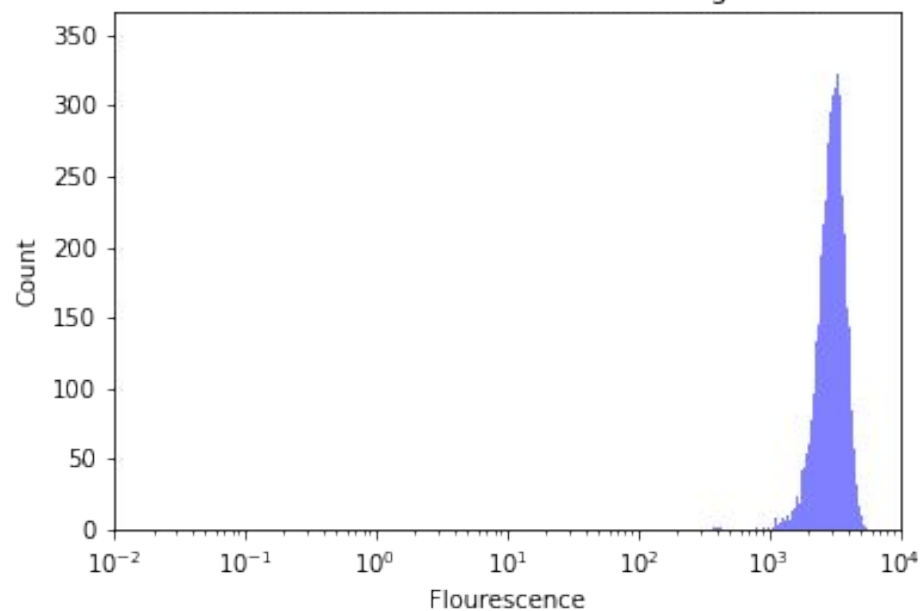
Example

Distribution for guide 0
 $\mu = 3197.981694297485$, $\sigma = 675.9654539290184$

Distribution of flourence (x axis linear)

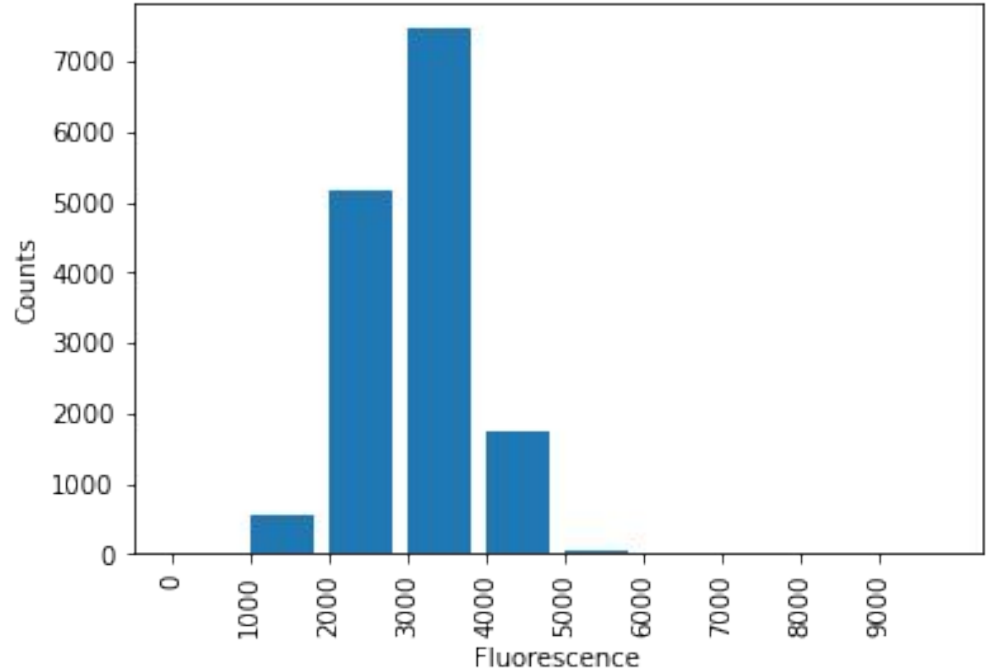


Distribution of flourence (x axis log scale)



Binning

- The cell sorter is not able to record the exact fluorescence provided by a cell
- Instead, it determines if that cell's fluorescence falls within a certain range
- The amount of resolution we expect the cell sorter to be able to distinguish can be determined by the user



Outputs

- Two data files output:

`mock_bins_params`: contains the true means and variances

	ID	Mean	Variance
0	poxBp_T7_TGT	3197.981694	675.965454
1	poxBp_T17_GAA	5767.974676	706.882804
2	poxBp_T18_TGA	5142.343422	854.816014
3	poxBp_T20_GAT	5963.503534	795.546748
4	poxBp_T21_AGA	4639.914297	785.172958
...
1727	J3_B195_CGC	4023.553637	796.059613
1728	J3_B197_CAA	4169.147612	638.928781
1729	J3_B201_TGC	5611.93888	773.825383
1730	J3_OT	5529.450726	703.203954
1731	J3_OT_mutTATA	6058.197958	686.367314

`mock_bins_counts`: contains the histograms for each guide

	ID	0	1000	2000	3000	4000	5000	6000	7000	8000	9000
0	poxBp_T7_TGT	6.0	559.0	5172.0	7461.0	1741.0	61.0	0.0	0.0	0.0	0.0
1	poxBp_T17_GAA	0.0	0.0	3.0	96.0	1938.0	7422.0	4942.0	586.0	13.0	0.0
2	poxBp_T18_TGA	0.0	1.0	93.0	1298.0	5064.0	6162.0	2148.0	230.0	4.0	0.0
3	poxBp_T20_GAT	0.0	0.0	1.0	93.0	1576.0	6170.0	5734.0	1354.0	69.0	3.0
4	poxBp_T21_AGA	0.0	9.0	274.0	2858.0	6992.0	4239.0	606.0	22.0	0.0	0.0
...
1727	J3_B195_CGC	0.0	83.0	1363.0	5877.0	6013.0	1569.0	94.0	1.0	0.0	0.0
1728	J3_B197_CAA	0.0	3.0	495.0	5398.0	7681.0	1393.0	29.0	1.0	0.0	0.0
1729	J3_B201_TGC	0.0	0.0	10.0	277.0	2954.0	7155.0	4063.0	526.0	15.0	0.0
1730	J3_OT	0.0	0.0	2.0	194.0	3134.0	7945.0	3457.0	262.0	6.0	0.0
1731	J3_OT_mutTATA	0.0	0.0	0.0	18.0	936.0	6037.0	6672.0	1292.0	45.0	0.0

What's in a fastq?

4 rows:

1. Sequence identifier, which contains the following elements:
 - a. @
 - b. instrument ID
 - c. run number on instrument
 - d. flowcell ID
 - e. lane number
 - f. tile number
 - g. x coordinate of cluster
 - h. y coordinate of cluster
 - i. UMI sequences for Read 1 and Read 2
 - j. read number (1)**
 - k. Y if the read is filtered, N otherwise
 - l. control number
 - m. index (2)**
- 2. sequence (3)**
3. Quality score identifier line (+)
4. Quality Score

All items in our data are placeholders **except** for

1. **read number:** will be 1 or 2 depending on if its read 1 or read 2
2. **index:** the barcode sequence associated with the bin
3. **sequence:** the sequence coming from the guide variant (different depending on if its read1 or read2)

Psuedocode

- For each guide variant from the mock_bins_counts file:
 - Generate read1
 - seq = this guide's sequence
 - seq_i = index in seq where TAGG is found
 - read1 = seq[seq_i : seq_i + 39]
 - Generate read2
 - rev_i = index in rev where CCTA is found
 - rev = this guide's reverse complement
 - read2 = rev[rev_i - 32: rev_i + 4]
 - For each bin
 - set the barcode to the barcode sequence associated with this bin
 - let n = number of entries in this bin
 - output n entries with read number = 1, index = barcode, and sequence = read1
 - output n entries with read number = 2, index = barcode, and sequence = read2

Example

- poxBp T7 TGT has the following sequence:

```
CTGAAGTCAGCCCCATACGATATAAGTTGTTACTAGATTGACAGCTAGCTCAG  
TCCTAGGTATAATACTAGTTAACGGTTAAATAGCCCGATGTTTTAGAGCTAGA  
AATAGCAAGTTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGT
```

Example - Calculate read1

CTGAAGTCAGCCCCATACGATATAAGTTGTTACTAGATTGACAGCTAGCTCAG
TCC TAGG TATAATACTAGTTAACGGTTAAATAGCCCGATGTTTTAGAGCTAGA
AATAGCAAGTTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGT

- find TAGG

Example - Calculate read1

CTGAAGTCAGCCCCATACGATATAAGTTGTTACTAGATTGACAGCTAGCTCAG
TCC TAGGTATAATACTAGTTAACGGTTAAATAGCCCGATGTTT TAGAGCTAGA
AATAGCAAGTTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGT

- find TAGG
- grab 39 base pairs, including and starting at TAGG

Example - Calculate read1

CTGAAGTCAGCCCCATACGATATAAGTTGTTACTAGATTGACAGCTAGCTCAG
TCC TAGGTATAATACTAGTTAACGGTTAAATAGCCCGATGTT TTAGAGCTAGA
AATAGCAAGTTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGT

- find TAGG
- grab 39 sequences
- read1 = TAGGTATAATACTAGTTAACGGTTAAATAGCCCGATGTT

Example - Calculate read2

ACTTTTTCAAGTTGATAACGGACTAGCCTTATTTTAACTTGCTATTTCTAGCTC
TAAACATCGGGCTATTTAACCGTTAACTAGTATTATACCTAGGACTGAGCTA
GCTGTCAATCTAGTAACAACCTTATATCGTATGGGGCTGACTTCAG

- Take reverse complement

Example - Calculate read2

ACTTTTTCAAGTTGATAACGGACTAGCCTTATTTTAACTTGCTATTTCTAGCTC
TAAACATCGGGCTATTTAACCGTTAACTAGTATTATACCTAGGACTGAGCTA
GCTGTCAATCTAGTAACAACCTTATATCGTATGGGGCTGACTTCAG

- Take reverse complement
- Find CCTA

Example - Calculate read2

ACTTTTTCAAGTTGATAACGGACTAGCCTTATTTTAACTTGCTATTTCTAGCTC
TAAAACATCGGGCTATTTAACCGTTAACTAGTATTATACCTAGGACTGAGCTA
GCTGTCAATCTAGTAACAACCTTATATCGTATGGGGCTGACTTCAG

- Take reverse complement
- Find CCTA
- Take 36 sequences, including and ending with CCTA

Example - Calculate read2

ACTTTTTCAAGTTGATAACGGACTAGCCTTATTTTAACTTGCTATTTCTAGCTC
TAAAACATCGGGCTATTTAACCGTTAACTAGTATTATACCTAGGACTGAGCTA
GCTGTCAATCTAGTAACAACCTTATATCGTATGGGGCTGACTTCAG

- Take reverse complement
- Find CCTA
- Take 36 sequences, ending with CCTA
- read2 is ATCGGGCTATTTAACCGTTAACTAGTATTATACCTA

Example

- Let's say that this poxBp T7 TGT has 32 reads in the second bin.
- The second bin's barcode is **AACCGCAA**.

So the 32 read1s look like:

```
@SIM:1:FCX:1:14:6329:1045:1:N:0:AACCGCAA
TAGGTATAATACTAGTTAACGGTTAAATAGCCCGATGTT
+
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

And the 32 read2s look like

```
@SIM:1:FCX:1:14:6329:1045:2:N:0:AACCGCAA
ATCGGGCTATTTAACGGTTAACTAGTATTATACCTA
+
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Results

- This process was repeated for all guide variants and each of the bins to get a total of 25,980,000 records per file.
 - Took 5 minutes to generate on my laptop
 - The fastq files ARE HUGE (3gbs each!!!!)
- I was able to read through the files with Biopythons SeqIO in the order of minutes, implying that processing the data with Biopython could be completed in a reasonable amount of time for researchers

Discussion

- This data was never confirmed to be a good stand-in for real world data,
 - real data was not generated to validate the synthetic data's format
- The Illumina sequencing and FACS would be too time consuming and tedious to set up.
 - Abandoned for another simpler sequencing method called Amplicon sequencing.

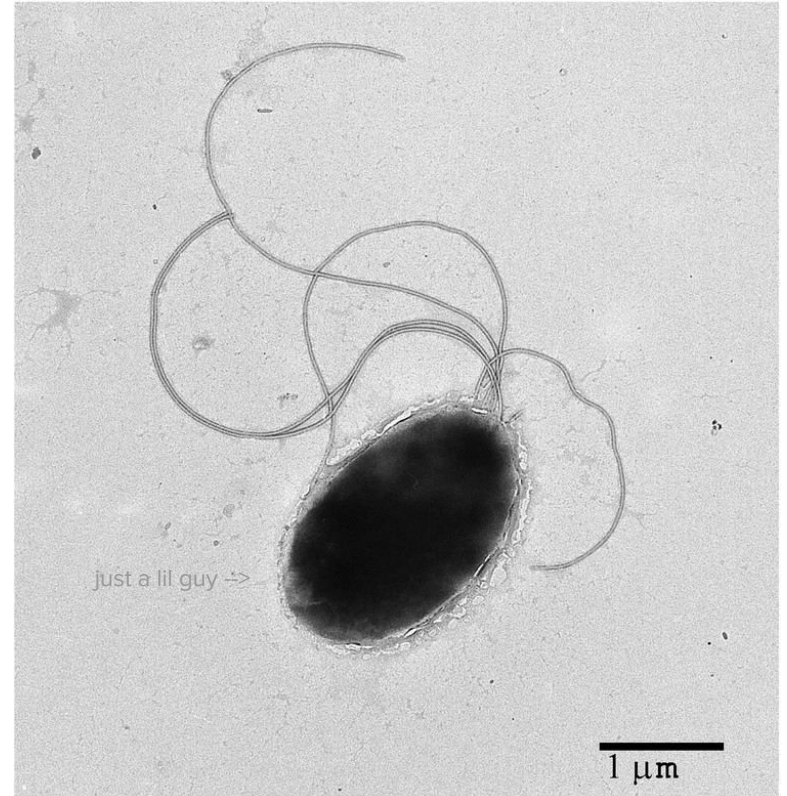
Future work for this project would involve gathering real world FACS Illumina data to confirm the validity of the mock data

Future work should also consider using the mock data to test analysis pipelines

Project 2: *P. putida* Gene Expression CRISPRa Filtering

P. putida

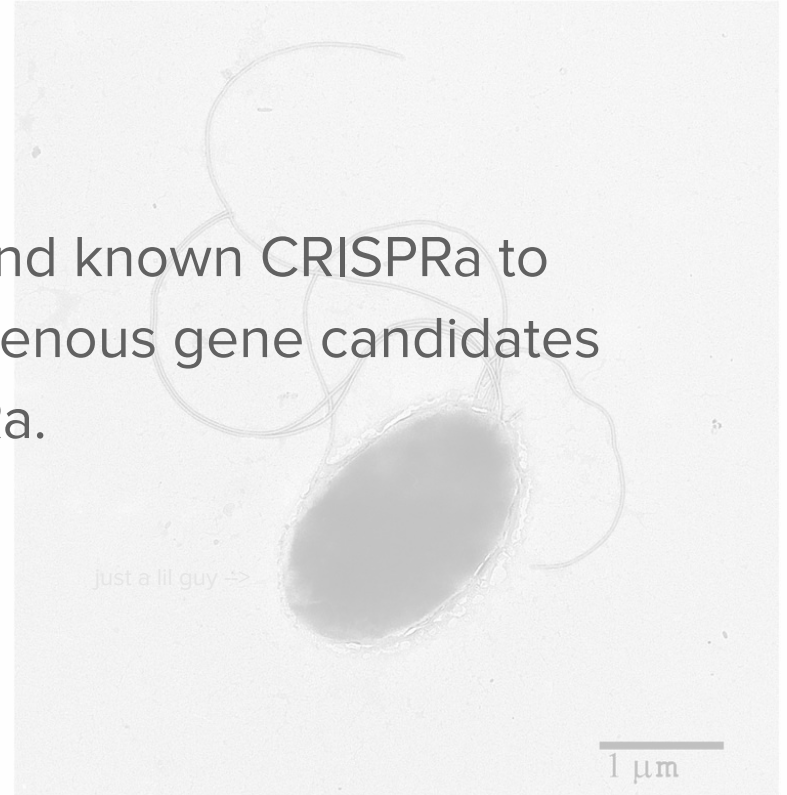
- Soil bacterium
- Potential to be a great vehicle to convert various renewable materials into desired compounds with high precision and efficiency
 - Example: fluorescent siderophore pyoverdine, which is used for applications such as plant growth promotion
- *P. putida* CRISPRa has been demonstrated, but the rules to suggest effective CRISPRa are not available
- We aimed to discover which endogenous genes were good candidate for CRISPRa



P. putida

- Saprotrophic soil bacterium
- potential to be a great vehicle to convert various renewable materials into desired compounds with high precision and efficiency
 - Example: fluorescent siderophore pyoverdine, which is used for applications such as plant growth promotion
- *P. putida* CRISPRa has been demonstrated, but the rules to suggest effective CRISPRa is not available :(
- We aimed to discover which endogenous genes were good candidate for CRISPRa.

Goal: Use *P. putida* datasets and known CRISPRa to generate a list of suitable endogenous gene candidates for CRISPRa.



Baseline Expression

- CRISPRa is sensitive to promoter strength/baseline expression
 - If a promoter is too weak, it will be difficult to activate and acquire high fold change.
 - If a promoter is too strong, fold-change due to activation is minimal due to other limiting factors.
- The range of an acceptable baseline is not known and must be inferred from existing data.

Data Sources

- RNA-Seq Data
 - Data that measures the baseline gene expression of various genes in *P. putida*.
 - Provided by Joshua Elmore
- List of Activated Genes
 - 11 genes were tested to see if they were suitable candidates for activation based on their baseline expression.
 - 4 of those genes were found to have suitable activation (PP 1776, PP 1992, PP 0786, PP 3668)
 - Generated by Ice (Cholpisit Kiattisewee)
- Genome Scale Model Outputs
 - This file contained predictions of effective CRISPRa/i genes from a genome-scale model
 - Produced by our collaborator Hector Garcia Martin
- Primary TSS Data
 - Indicates which genes are TSS primary promoters

Data Sources

- RNA-Seq Data

- Data that measures the baseline gene expression of various genes in *P. putida*.
- Provided by Joshua Elmore

- List of Activated Genes

- 11 genes were tested to see if they were suitable candidates for activation based on their baseline expression.
- 4 of those genes were found to have suitable activation (PP 1776, PP 1992, PP 0786, PP 3668)
- Generated by Ice (Cholpisit Kiattisewee)

- Genome Scale Model Outputs

- This file contained predictions of effective CRISPRa/i genes from a genome-scale model
- Produced by our collaborator Hector Garcia Martin















- Primary TSS Data

- Indicates which genes are TSS primary promoters

RNA Seq Data

- Data that measures the baseline gene expression of various genes in *P. putida*.
- Provided by Joshua Elmore
- Over 5000 genes
- 4 columns each:
 - locus tag.
 - An identifier that represents the gene. By convention, it begins with "PP" followed by an underscore and then a 4 letter
 - name
 - A description of the gene's product.
 - Raw Read.
 - The raw data read for this particular gene.
 - FPKM.
 - Short for Fragments Per Kilobase of transcript per Million mapped reads.
 - Basically, this is the total reads for that sample, divided by 1 million, divided by the length of the gene.
 - This was used in the pipeline to represent baseline gene expression

- Consists of 14 CSVs
- 4 different experimental conditions
 - 2 different promoters
 - 2 different nitrogen sources (ammonium NH_4 or nitrate NO_3)
- 3 - 4 replicates per experiment

	GSM4430334_JE1657-GNH4-1.csv
	GSM4430335_JE1657-GNH4-2.csv
	GSM4430336_JE1657-GNH4-3.csv
	GSM4430337_JE1657-GNH4-4.csv
	GSM4430338_JE1657-GNO3-1.csv
	GSM4430339_JE1657-GNO3-2.csv
	GSM4430340_JE1657-GNO3-3.csv
	GSM4430341_JE1657-GNO3-4.csv
	GSM4430342_JE2212-GNH4-1.csv
	GSM4430343_JE2212-GNH4-2.csv
	GSM4430344_JE2212-GNH4-3.csv
	GSM4430345_JE2212-GNO3-1.csv
	GSM4430346_JE2212-GNO3-2.csv
	GSM4430347_JE2212-GNO3-3.csv















RNA Seq Data

- Data that measures the baseline gene expression of various genes in *P. putida*.
- Provided by Joshua Elmore
- Over 5000 genes
- 4 columns
 - **locus tag.**
 - An identifier that represents the gene. By convention, it begins with "PP" followed by an underscore and then a 4 letter id
 - **name**
 - A description of the gene's product.
 - **Raw Read.**
 - The raw data read for this particular gene.
 - **FPKM**
 - Short for Fragments Per Kilobase of transcript per Million mapped reads.
 - Basically, this is the total reads for that sample, divided by 1 million, divided by the length of the gene.
 - This was used in the pipeline to represent baseline gene expression

RNA Seq Data

- Consists of 14 CSVs
- 4 different experimental conditions
 - 2 different promoters
 - 2 different nitrogen sources (ammonium NH4 or nitrate NO3)
- 3 - 4 replicates per experiment

- Nitrogen sources
 - The experiment with ammonium will be closer to what we plan to perform
 - So the user can specify to use only these datasets in the filtering

	GSM4430334_JE1657-GNH4-1.csv
	GSM4430335_JE1657-GNH4-2.csv
	GSM4430336_JE1657-GNH4-3.csv
	GSM4430337_JE1657-GNH4-4.csv
	GSM4430338_JE1657-GNO3-1.csv
	GSM4430339_JE1657-GNO3-2.csv
	GSM4430340_JE1657-GNO3-3.csv
	GSM4430341_JE1657-GNO3-4.csv
	GSM4430342_JE2212-GNH4-1.csv
	GSM4430343_JE2212-GNH4-2.csv
	GSM4430344_JE2212-GNH4-3.csv
	GSM4430345_JE2212-GNO3-1.csv
	GSM4430346_JE2212-GNO3-2.csv
	GSM4430347_JE2212-GNO3-3.csv

List of Activated Genes

- 11 genes were tested to see if they were suitable candidates for activation based on their baseline expression.
- 4 of those genes were found to have suitable activation (PP 1776, PP 1992, PP 0786, PP 3668)
- Generated by Ice (Cholpisit Kiattisewee)

locus tag
PP 1776
PP 4812
PP 3839
PP 1992
PP 0786
PP 1972
PP 3668
PP 5046
PP 1231
PP 4701
PP 3161

1. Normalization

The FPKM was normalized by the FPKM of a reference gene

- Pseudocode:
 - For each of the RNA Seq Files
 - Get the FPKM of npII in this file
 - For each gene in the file:
 - Divide its FPKM by the FPKM of npII

2. Averaging Across Results

Average the datasets, so there is only 1 dataset per experimental condition

- Pseudocode:
 - For each experimental condition:
 - For each replicate in that experimental condition, average the FPKM of the datasets

- There are 4 different experimental conditions:
 - (1) JE1657 NH4, (2) JE1657 NO3, (3) JE2212 NH4, (4) JE2212 NO3,
 - so we should have 4 datasets now

3. Filter by Primary TSS

Remove any genes not in the primary TSS dataset

Filtering by Activation

- Filter genes that have activation within a desirable range
 - Found by looking at the FPKMs of the genes considered to have suitable activation: PP 1776, PP 1992, PP 0786, PP 3668
- A question emerged: how do we do this when we have multiple datasets?
 - Do we filter separately for each dataset?
 - Do we filter by a global maximum and minimum?
 - If a gene is filtered out of one dataset, but not another, is it still considered a suitable candidate, or does it need to not be filtered out by any dataset?

locus tag
PP 1776
PP 4812
PP 3839
PP 1992
PP 0786
PP 1972
PP 3668
PP 5046
PP 1231
PP 4701
PP 3161

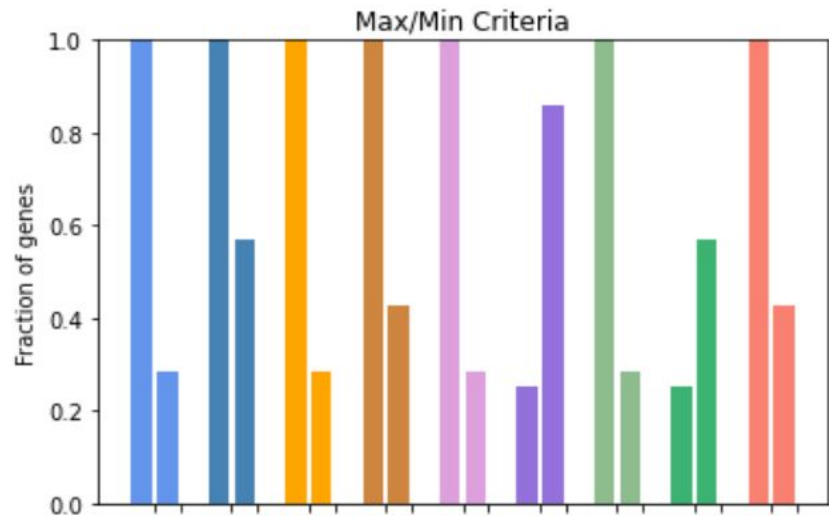
Data Analysis Experiments

In order to figure out which method was the best, an experiment was performed:

- For each filtering method
 - Count how many of the 4 genes considered suitable candidates were kept in the dataset
 - Count how many of the 7 genes considered not suitable were thrown out of the dataset
- The filtering method that kept the most suitable genes and threw out the most unsuitable genes is our winner!

Experiment Results

- Basically we want both bars to be high
- Winner:
 - Using a maximum and minimum for each dataset, and requiring the gene to be present in each dataset after filtering.
 - Kept all 4 genes considered suitable candidates
 - Threw out 4 of the 7 genes considered not suitable



Left Bars: Fraction of genes that should be in the set are present
Right Bars: Fraction genes that aren't in the set aren't present

Results

- At this point, the software reduced 5571 genes down to 503 genes, which is 9% of input genes.
- For CRISPRa, 15 genes were provided by the genome scale model
- 5 of these 15 genes were also output by my software:
 - PP_2082, PP_1830, PP_1075, PP_4965, PP_1972
- The pipeline finishes running in under one minute
 - Reasonable for this type of analysis

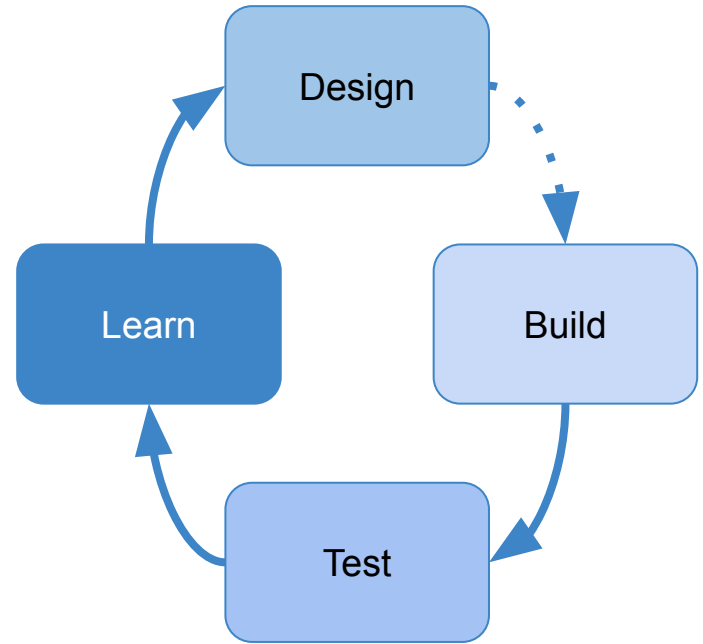
Discussion

- Work meant to be part of the Learn and Design part of Design-Build-Test-Learn cycle
 - It's not a cycle if we don't go back to the build phase!
 - The genes selected after filtering have not been built nor tested to validate these findings
 - Future work should continue the cycle and test the validity of the approach and reassess and design the software to incorporate new insights.
- Another large limitation of this project is the validity of the filtering analysis method.
 - It's not necessarily founded in statistics
 - Hard to verify without any validation data

Conclusion

Limitations

- Biggest limitation: You can't have a DBTL cycle if the cycle doesn't continue
- Future work must validate these approaches so analysis can be validated and learned from



Acknowledgments

- James, thank you so much for teaching introduction to synthetic biology, introducing me to this field, and inviting me to join the lab!
- Ice and Ben for being amazing mentors throughout the years
 - Ice especially for mentoring me through the *P. putida* project!!!
- Ava and Ryan for providing me with resources for this thesis
- Niel for generating the barcode sequences for the mock data generation project
- Jason Fontana for getting me started with CRISPRa scoring function work
- Dennis, for reading my thesis and giving me a non-biology & non-CSE perspective
- Matt, for always encouraging me and also helping me with the blue light projector
- My parents, Sheri and Craig, for buying me a computer, always being there for me, encouraging me, supporting me throughout my whole life! <3
- The rest of my friends and family for supporting me through the process

I wouldn't be where I am today without you!



Questions

Thank you so much!

Contact me post-graduation at joelynelson3333@gmail.com