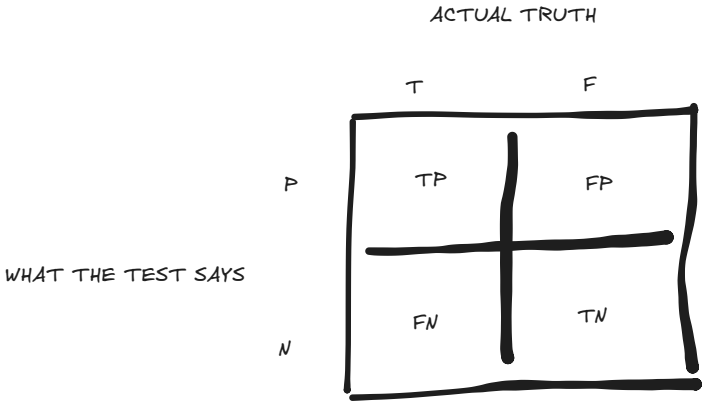
MPRA power analysis

Overview: error control

This section gives a high-level overview of how we will control error rates. I will handwave the mathematical details here, then go over them in a little more detail in subsequent sections.

We often collect data to try to answer specific questions about biological phenomena. However, just looking at the data is usually not sufficient to judge whether the data point towards one answer over the other. In that case, we want some kind of criterion, or "test", to tell us what the data suggest. If we restrict ourselves to yes/no questions (e.g. "Is the presence of this CRE changing expression?" or "Is the presence of this variant changing expression?") then we can consider a couple of possible outcomes. Either the answer to the question is actually "true" or actually "false", and either the test reports that it is "true" or that it is "false".



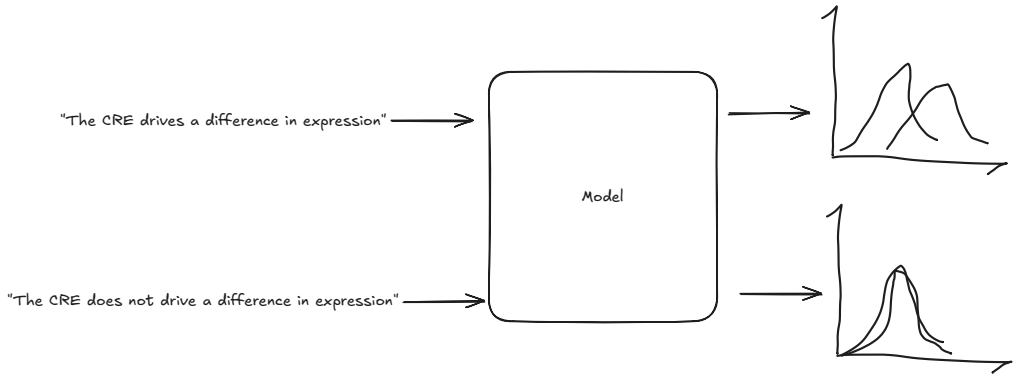
Of the four possibilities, two are desirable (true positive, true negative) and two are undesirable (false positive, false negative). Importantly, we only have access to what our test tells us (positive / negative) not the actual truth : otherwise we would not have to run any test at all! We wish then to assign some kind of numbers to each of the undesirable cases. If we can't know exactly, at least we can quantify exactly how bad the two problems are. These are the false positive rates, and the false negative rates.

Having some estimate of these rates is generally desirable for a couple reasons. First, it helps us become appropriately confident in our conclusions. (e.g. we should be more skeptical of a positive result when the false positive rate is high than when it is low).

Second, measuring a problem is often a good first step to ameliorating it, too. We may make decisions about how we design our experiment with the goal of reducing the rate of the undesirable cases.

How can we quantify these cases, given that we don't have access to real, ground-truth reality? Simple! We invent a model where we DO know the ground-truth! Because we made it up! Then, as long as our model corresponds to reality then the estimates of the undesirable cases will also correspond to reality!

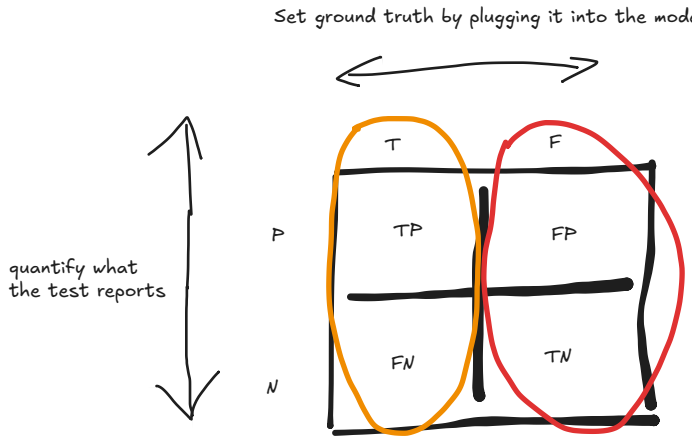
What do I mean by a "model" here? These can take a lot of forms, but generally what I am referring to is some equation or computer program that does the following: you start with a possible answer to your question e.g. "the CRE does NOT drive additional expression" or "the CRE increases expression by ten transcripts", and the model shows you *what your data would look like in that case*.



Once you have such a model you can ask questions that directly quantify the two undesirable cases.

Measuring false-positives with p-values

For example, suppose you have performed an actual experiment, and the test has returned "positive". How surprising would that result be if there were really no effect? Formally, we ask this question as "given that there is no effect, what is the probability that we would observe a result as extreme or more extreme than the one we observed?" To answer this question, you plug in the ground-truth "there was actually no effect" into your model, get "here is what the data would look like if there was really no effect" and then see how frequently you would get positives.



The case I just described has you plug in "there was no effect". So essentially, you are setting ground truth to "False" and then looking at FP and TN : this corresponds to the red circle above. In this way, you quantify one of the two undesirable cases (FP). The probability I described ("given that there is no effect, what is the probability that we would observe a result as extreme or more extreme than the one we observed?") is of course the p-value! Note what it is conditioned on: it tells you how often *an experiment with no effect* would produce a result like yours, not how likely it is that *your* particular positive is false.

Some investigators like to set a threshold for a false-positive probability they are comfortable with in advance. For example, we might say that, in cases where there is really no effect, we are comfortable calling a positive 5% of the time. We might call this an "alpha" value, and check it against our p value. Then, if $p<0.05$ we are satisfied, and if $p>= 0.05$, we are not satisfied. (Personally, I much prefer reporting raw p-values, and find the whole concept of statistical significance slightly suspect.)

Measuring false-negatives with power analysis

What about the other undesirable case: the false negative rate? We can handle it in much the same way, corresponding to the orange circle above. Here, we take some ground truth like "the CRE increases expression by ten transcripts", plug it into the model, and see what the data look like. Then, we compute the probability that our test fails to find the effect, and this number is the false negative rate!

One additional complication: oftentimes tests have an easier time detecting larger effects. So the false negative rate for detecting a change of 10 transcripts will probably be higher than the false negative rate for detecting a change of 50 transcripts. For this reason, you may want to choose the effect size you plug into your model based on the minimum effect

size you want to be able to detect. Alternatively, you can put lots of different possible changes as ground truth into your model, and plot how the false negative rate changes as a function of effect size:

This is unlike false positive control, which can be conveniently summarized by a single number.

As you may have guessed, this is a power analysis! Statistical power is just 1 minus our false negative rate (so a design that misses a given effect 20% of the time has 80% power to detect it).

There are lots of ways to make the models I described.

1. Sometimes these models are analytical. That is, they are closed-form equations, where, given some assumptions about the shape of the data, you can compute the false positive and negative rates exactly.
2. Sometimes these models use empirical data directly. These are approaches like bootstrap resampling and permutation tests.
3. Sometimes these models are based on simulation, and you simulate what the data would look like based on some prior idea about what it should look like.

Personally, I am partial to simulation-based approaches because they work in a very broad set of cases (allowing you to include all kinds of weird properties of, say, MPRA data), and require fewer assumptions about the structure of the data. Their main disadvantage is that they can be computationally intensive. But computers are very fast now, so with a bit of clever programming this disadvantage doesn't really matter that much. I will focus on simulation-based approaches throughout.

Note about rates

The false positive rate is the rate among all the cases where there is really no effect; the false negative rate is the rate among all the cases where there really is one. This considers each separately, and does not measure the *absolute number* of each. To convert these rates into a number, you need to estimate *what fraction of the tests you run have a real effect*. (e.g. of 200 experiments where half have a real effect, 100 have no real effect. At a 5% false positive rate, about 5 of those 100 will come back positive.)

This also means that "what fraction of my positives are wrong?" is NOT the false positive rate, and depends on that same fraction. E.g. if real effects are 0.1% of the tests, a 5% false positive rate can still leave you with a hit list that is mostly false positives.

Random variables and probability mass functions

Now I will talk a little bit about some mathematical objects which will be useful in constructing the models I mentioned.

First, the *random variable*. You are doubtless familiar with normal variables, e.g. $x = 3$. A random variable is much like a normal variable, except that instead of taking a *single* value (3), it takes a *variety* of values with different probabilities. Where a normal variable is good at describing some concrete quantity (number of bananas, how many meters long a road is, etc) *random* variables are useful for describing cases that involve some uncertainty e.g. "how many people will enter my store in the next hour" . The outcomes of scientific experiments often fall into the latter category.

What exactly do I mean by "probabilities"? There are actually a variety of ways we could define this formally, but for simplicity's sake we will adopt an informal, commonsense definition. A probability is some number from 0 and 1 which reflects how often an event occurs. So a probability of 0.5 means the event happens half of the time. A probability of 0.1 means that the event occurs 10% of the time, etc. Note that this definition of probability is very similar to "frequency of event/total trials".

We will use the following notation to denote probabilities:

$$P(\text{some event}) = \text{probability that the event occurs}$$

A probability mass function is the mathematical object which *defines* a random variable. The probabilities of a probability mass function sum to 1 (since SOME outcome must occur).

This is getting a bit abstract, so let's go through a specific example. Suppose the 'experiment' we wish to model is rolling a six-sided die. We might then summarize the possible outcomes as follows:

$$P(\text{die rolls a } 1) = 1/6$$

$$P(\text{die rolls a } 2) = 1/6$$

$$\dots$$

$$P(\text{die rolls a } 6) = 1/6$$

As you can see, the die is fair, and each outcome has a one in six chance of occurring. Let us now describe the experiment formally.

We model the six-sided-die-experiment as a random variable X, with the probability mass function

$$p_X(x)$$

Which we define as

$$p_X(x) = P(X = x)$$

Where

$$p_X(x) = \begin{cases} 1/6 & \text{if } x = 1 \\ 1/6 & \text{if } x = 2 \\ 1/6 & \text{if } x = 3 \\ 1/6 & \text{if } x = 4 \\ 1/6 & \text{if } x = 5 \\ 1/6 & \text{if } x = 6 \\ 0 & \text{otherwise} \end{cases}$$

Essentially then, a probability mass function is one which, for some random variable, takes some outcome as its sole argument, and returns the probability of that outcome occurring.

(The notation can get a tad confusing: Little x is some value which the die can report : 1, 2, 3... Big X is the random variable itself. p_X is the probability mass function of the random variable X. Big P is the probability of some event. To add to the confusion, there are multiple sets of notation in use by different statisticians! I follow the example of *statlect*, my personal favorite.)

The term "distribution" is often used to loosely refer to probability mass functions.

The case I showed above is very simple. You might reasonably wonder why we need all this fancy notation at all. Why "define a function" when you could just list every outcome and its probability? The answer is that the notation becomes more useful when things get more complicated! The inside of the brackets doesn't have to be a mere enumeration of all possible outcomes and their probabilities, it can be an equation. For example, if a random variable Y follows the binomial distribution, we say that its probability mass function is:

$$p_Y(k) = P(Y = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

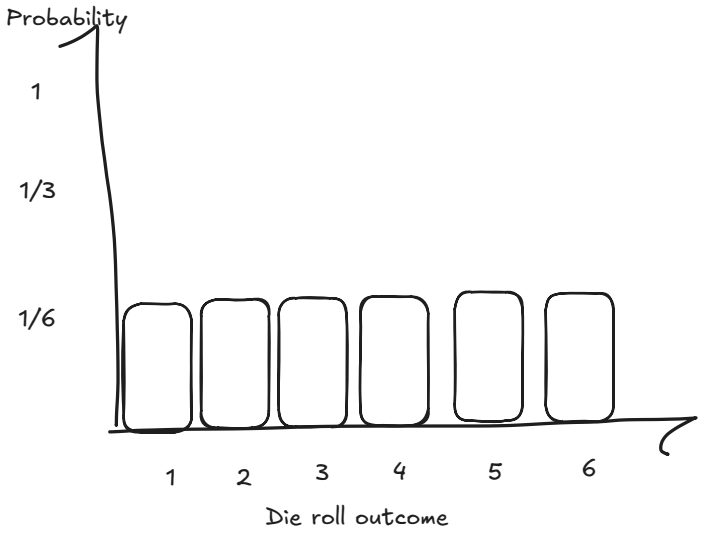
If we know certain parameters (n and p) we can plug in k , and get the probability that the random variable Y assumes value k. Don't worry about the particulars here: we will not discuss the binomial distribution further. I just present it as an example of why we want to define this notation.

Aside about probability density

The probability mass function is only defined for random variables which have *discrete* outcomes. e.g. the die can roll 3 or 4 or 5 but never 4.5, or 5.6 or 3.14159.... Sometimes, you want to model an experiment that has a continuous outcome (e.g. after drug treatment, the tumor shrinks to Y grams. We wish to model Y.) In this case, you use a *continuous* random variable with a probability *density* function. However, most of the equations are the same (you just replace sums with integrals) and the data we are interested in, read-count data, are discrete, and not continuous. So I will ignore continuous random variables and probability density functions here. But it is good to know they exist.

Plotting probability mass functions

I often find it helpful to plot probability mass functions as histograms, to examine how they behave visually. I put the value assumed by the variable on the x axis, and the probability of occurring on the y axis. Here is an example of how such a plot would look for our die-rolling experiment earlier.



Expected value

(This is a synonym of "mean", just defined formally and for a probability mass function.) The expected value of a random variable is denoted with a big $E[\cdot]$, and defined as

$$E[X] = \sum x \cdot p_X(x)$$

For all valid x that the pmf can take.

In the case of our die-rolling experiment from earlier, this becomes:

$$E[X] = 1 \cdot 1/6 + 2 \cdot 1/6 + 3 \cdot 1/6 + 4 \cdot 1/6 + 5 \cdot 1/6 + 6 \cdot 1/6$$

$$E[X] = (1/6)(1 + 2 + 3 + 4 + 5 + 6)$$

$$E[X] = (1/6)(21)$$

$$E[X] = 3.5$$

This means that the mean, or expected value of the distribution is 3.5.

Variance

The variance of a random variable X is defined as:

$$Var[X] = E[(X - E[X])^2]$$

The more the data are spread out around $E[X]$, the higher the variance. The less the data are spread out, the lower the variance. The standard deviation is the square root of the variance.

Using probability mass functions to compute the probability of more complicated events

Suppose we have our die-rolling experiment from before, and we want to calculate a more complicated probability. Say, $P(X > 3)$. In english, we would say that this is "the probability of rolling a number higher than 3". We can compute this as:

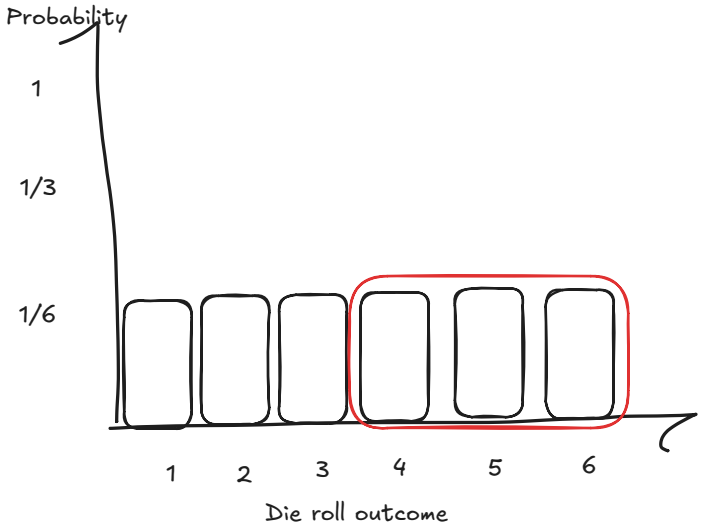
$$P(X > 3) = p_X(4) + p_X(5) + p_X(6)$$

$$P(X > 3) = 1/6 + 1/6 + 1/6$$

$$P(X > 3) = 1/2$$

Unsurprisingly, the chance is 50%.

You can also consider this in terms of the plots:



You can find $P(X > 3)$ by taking the AREA under the probability mass function curve corresponding to the results you are calculating the probability for. In this case, the x-dimension of the red square is 3 units, and the y dimension of the red square is $1/6$. So we say that the total area is $3 \cdot 1/6 = 3/6 = 1/2$: the same answer!

You can imagine that, if we have a probability mass function modeling an actual experiment, this kind of operation might be useful for answering questions like "what is the probability of getting a result this extreme or more extreme than 3..." like we posed earlier.

Drawing and fitting

Drawing and fitting are opposite operations.

"Drawing" from a distribution means getting specific events from a probability mass function. e.g. a series of 10 draws from our die example from earlier might look like:

```
>>> import random
>>> for i in range(10):
...     print(random.randint(1,6))
...
3
1
4
4
4
1
6
4
```

```
2
6
>>>
```

Each of the numbers (3, 1, 4, 4 ...) is a draw from the distribution.

As you sample more and more times, the mean and variance of the sampled data will approach the mean and variance of the underlying distribution.

"Fitting" takes some data (like the list of numbers, 3, 1, 4, 4 ..., as above) and *gives you a probability mass function which might have generated those data*. There are several techniques to do this, which I will not explain. Suffice it to say that it is possible.

The negative binomial distribution

There are many, many probability mass functions, each good for describing different kinds of data. The negative binomial distribution has proven useful for describing many aspects of read-count based sequencing data, so I will briefly cover it here.

There are various theoretical reasons to adopt NB, as it arises as a gamma-Poisson mixture, but for our brief treatment here, suffice it to say that it works well empirically.

nb1, nb2, and nbp

NB has two parameters: a mean and a dispersion. What distinguishes the common parameterizations is how the variance relates to the mean μ:

- **nb1**: $Var = \mu + \alpha\mu$. Variance is a constant multiple of the mean.
- **nb2**: $Var = \mu + \alpha\mu^2$. Variance grows quadratically. This is what DESeq2 and edgeR use.
- **nbp**: $Var = \mu + \alpha\mu^p$, with the exponent p estimated from the data rather than fixed at 1 or 2.

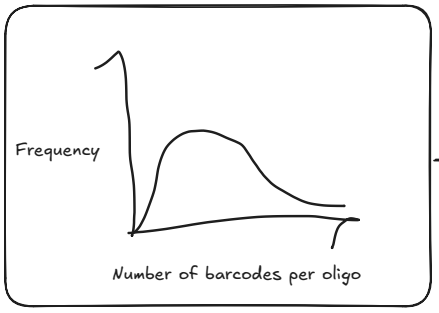
nb2 is traditional, and works well in most cases in my experience.

Example of drawing from distributions to simulate an MPRA experiment

We would like to simulate MPRA data to perform error control. How should we do this? There are various approaches for simulating in the literature you can read about, so I will just briefly outline what a scheme would look like (at a fundamental level, this is the kind of scheme that I use).

Some CRE, with some activity

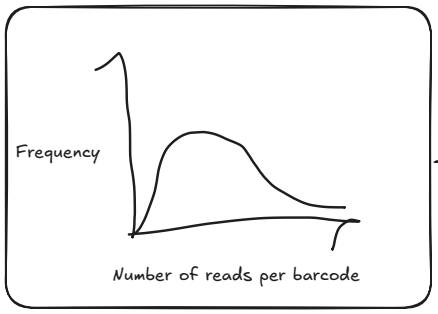
"How many barcodes was the CRE cloned into?"



Draw!

Some number of barcodes

How abundant is each barcode?

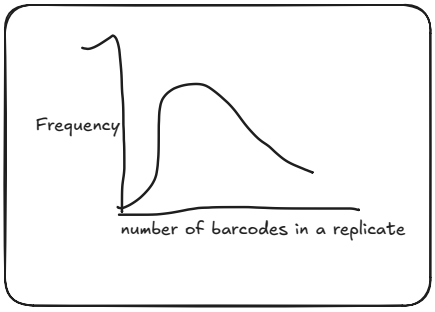


Draw!

some number barcodes at some abundance

Simulating the cloning

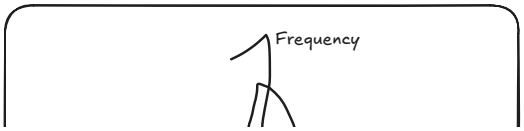
"Which barcode is transfected into which replicate?"



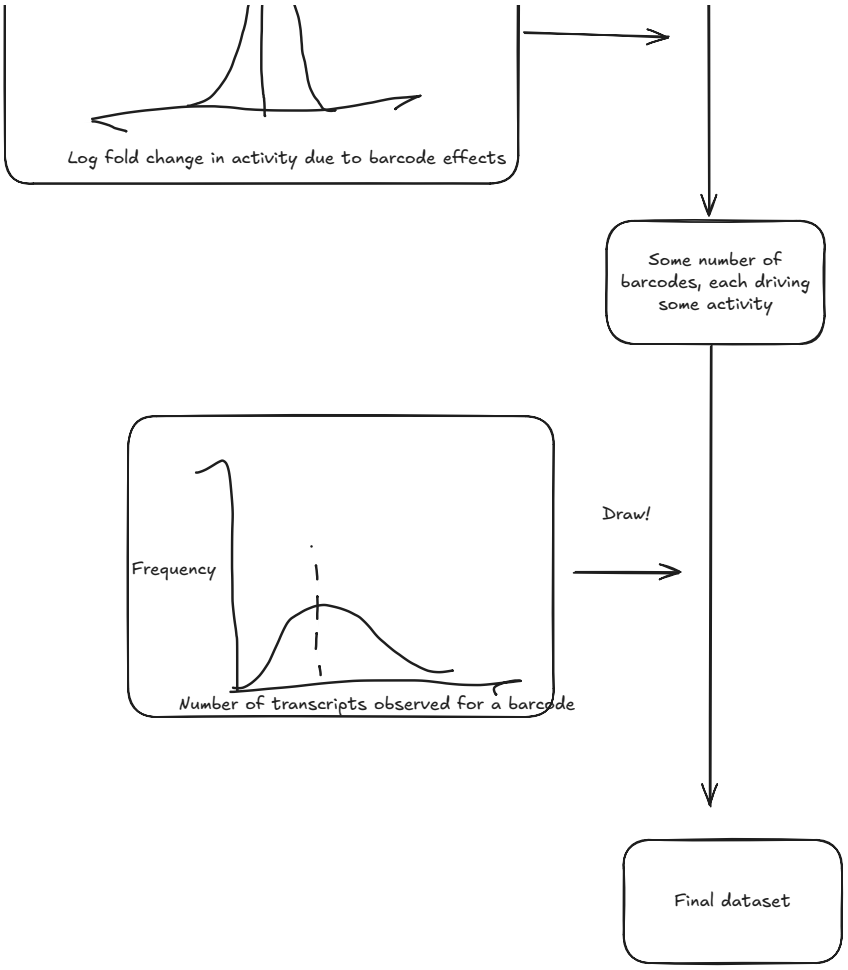
Draw x number of replicates!

Simulating the transfection

Some number of barcodes, transfected / not transfected



Draw!



As you can see, we simulate each of the steps of a real MPRA (cloning the library, introducing it into cells, and then transcription) by drawing from distributions. Most of these are well-described by some kind of NB2 in my experience, with a few exceptions. (e.g. barcode effects may be normally distributed on the log scale instead, since they are not the result of a counting process.)

In the end, you will get synthetic data:

		Replicate 1	Replicate 2	Replicate 3
CRE A	Barcode 1	62	71	55
	Barcode 2	48	39	52
	Barcode 3	83	95	77
CRE B	Barcode 4	151	168	143
	Barcode 5	129	117	138
	Barcode 6	194	172	205

Comparing synthetic data to ground-truth to estimate error rates

At the end, you have two things: the *actual ground truth* : these are the activities you put in at the beginning of the simulation. And the *simulated data*. Now, you can perform whatever statistical test you like to test whatever hypothesis you would like! You can then compare *what the test reports* to *what the ground truth actually was*. Some examples:

- If you simulate ten pairs of CREs with the same activity, and your test reports that 1 of the pairs is actually different, that's a false positive rate of 10%
- If you simulate 10 pairs of CREs with systematically varying activities (e.g. 1 transcript, 5 transcripts, 100 transcripts... etc) you can plot the "false negative rate as a function of fold change" curve I showed above. Note that because of the randomness of simulation, you will probably want to create and summarize a large number of simulated replicates.

Examining how error rates depend on design decisions

As you can see, at each point in the simulation you can make different *design decisions*: How many replicates? How many barcodes? How active are the CREs? How much do barcode-effects change transcription? If you put in all the parameters that produced some empirical dataset you should get data that look a lot like that empirical dataset! Then, if you want to determine the effect of these various design decisions on the error rates, you simply nudge one of them (e.g. change 5 replicates to 4 replicates) and measure the error rates again.