Modifying the Batch Size Calculation for False Positives and False Negatives.

The likelihood that a given sample is negative can be calculated as

Where *pos* is the percent of the sample population that is positive. This value can be determined by randomly testing from the samples collected.

My python code calculates the possibility of a single batch to be negative as: (see line 62)

In this equation *k* is the batch size. This represents the likelihood that a given batch contains no positive samples. Of note: This does not actually describe whether the batch comes back positive or negative. It is a measure of the actual likelihood that the batch contains at least one positive sample.

If that is the likelihood of a batch containing no positive samples, then the likelihood of a batch containing at least one positive sample is:

The expected number of tests per batch is then given as:

If you’re familiar with statistics, this number is the expectation value for a batch.

That assumes that each batch is tested once, and if a positive result comes back, each sample in the batch is retested individually.

How do we account for false positives and false negatives? While false positives and false negatives for individual samples might have severe consequences, neither one affects how many tests you have to perform, assuming no sample is retested. (Individuals could be retested, but those count as new samples.)

However, batch false positives and batch false negatives *do* affect how many tests must be performed. If we receive a false positive for a batch, we end up performing extra tests on individuals who don’t need it. If we receive a false negative for a batch, we do not test the individual samples, even when that batch contains an actual positive sample. If we let *fn* be the false negative rate for a batch and *fp* be the false positive rate, we modify the equations as follows:

Here, the expectation value is reduced by the likelihood of false negatives, and increased by the likelihood of false positives. (for sanity sake, it is easy to confirm that if *fn* and *fp* are 0, it reduces to the equation above.)