

Contents

- Heterogeneity in viral infection expected und
  - Overview of model assumptions
  - Detailed model
  - Defining the model parameters
  - The expected distributions for each sample
  - Conclusions

Table of Contents

- Heterogeneity in viral infection expected under simple Poisson model
  - Overview of model assumptions
  - Detailed model
  - Defining the model parameters
  - The expected distributions for each sample under the simple model
  - Conclusions

Heterogeneity in viral infection expected under simple Poisson model

This Jupyter notebook contains an analysis of the heterogeneity expected under some simple assumptions for the paper "Extreme heterogeneity of influenza virus infection in single cells" by Alistair Russell, Cole Trapnell, and Jesse Bloom.

Overview of model assumptions

We build a simple Poisson model for the expected heterogeneity under the following assumptions:

- Infection is a Poisson process
- Cells infected with a single virion with the full RNP express a given number of viral mRNAs, and cells infected with a single virion lacking an RNP gene express another number of viral mRNAs.
- Cells that are co-infected simply express a number of viral mRNAs proportional to the number of infecting virions. For instance, doubly infected cells express twice as much viral mRNA as singly infected cells.
- Each cell also has a fixed amount of cellular mRNA, and the amount of viral mRNA is added to this.

Note that this is a simple model that **we do not think accurately reflects reality**. The goal of this model is to determine how much heterogeneity we expect under the simplest scenario.

Detailed model

Let a fraction  $x_{full}$  of the virions have the full vRNP, implying that fraction  $1 - x_{full}$  lack an RNP gene.

Let the  $\langle v \rangle$  be the mean number of virions of any type (full or partial RNP) that infects the typical cell. Since we assume infection is Poisson, the distribution of the number of full-RNP virions  $v_{full}$  per cell is

$$\Pr(v_{full}) = e^{-(\langle v \rangle x_{full})} \times \frac{(\langle v \rangle x_{full})^{v_{full}}}{v_{full}!}$$

and the distribution of the number of partial-RNP virions  $v_{partial}$  per cell is

$$\Pr(v_{partial}) = e^{-(\langle v \rangle (1-x_{full}))} \times \frac{(\langle v \rangle (1-x_{full}))^{v_{partial}}}{v_{partial}!}.$$

Since infection is a Poisson process, the probability that a cell is infected by  $v_{full}$  full-RNP virions and  $v_{partial}$  partial-RNP virions is simply

$$\Pr(v_{full}, v_{partial}) = \Pr(v_{full}) \times \Pr(v_{partial}).$$

Let each cell have  $n_{cell}$  cellular mRNAs. Let each full-RNP virion add  $n_{full}$  viral mRNAs, and let each partial-RNP virion add  $n_{partial}$  viral mRNAs. The overall fraction  $f_{viral}$  of mRNA from virus cell if a cell is infected by  $v_{full}$  full-RNP virions and  $v_{partial}$  partial-RNP virions is simply

$$f_{viral} = \frac{v_{full}n_{full} + v_{partial}n_{partial}}{n_{cell} + v_{full}n_{full} + v_{partial}n_{partial}}.$$

We then simply want to determine the distribution of  $f_{viral}$  across cells among all cells that are infected (have  $v_{full} + v_{partial} > 0$ ).

Below we define the functions described above:

```
In [1]: import math

def f_viral(n_cell, n_full, n_partial, v_full, v_partial):
    """Total fraction of mRNA from virus"""
    numerator = (v_full * n_full + v_partial * n_partial)
    return numerator / float(numerator + n_cell)

def Pr_virions(v_mean, x_full, v_full, v_partial):
    """Probability `v_full` and `v_partial` full and partial virions infect cell"""
    full_mean = v_mean * x_full
    pr_full = math.exp(-full_mean) * (full_mean)**v_full / math.factorial(v_full)
    part_mean = v_mean * (1 - x_full)
    pr_part = math.exp(-part_mean) * (part_mean)**v_partial / math.factorial(v_partial)
    return pr_full * pr_part
```

Defining the model parameters

We define the model parameters based on the data in our experiments. Most of these parameters we define separately for each of the three samples (10hr, 8hr, and 8hr-2) that have substantial amounts of infection. The actual parameters are in the cell below.

First, we estimate  $\langle v \rangle$ . In our experiments, among the cells that are infected, we observe about 10% of them to have mixed viral barcodes. Our mixed barcodes will detect about half the co-infections, so we estimate that infected cells have about a 20% chance of being co-infected. Based on this, we estimate  $\langle v \rangle = 0.2$ , which means that the typical cell gets about 0.2 virions (this is essentially the estimate by assuming that all cells are either singly or doubly infected, and neglecting higher-order infections, which is reasonable since most cells are not multiply infected). Note that this estimate is not entirely consistent with the fraction of cells that exhibit any evidence of infection, indicating the assumption of Poisson infection is unlikely to be fully true. But we are building the model under that assumption.

We define  $x_{full}$  differently for each sample based on the observed fraction of infected cells with full RNP complements.

To set  $n_{cell}$ ,  $n_{full}$ , and  $n_{partial}$ , note that ultimately the units on these does not matter as they are simply used to calculate the fraction  $f_{viral}$ . So we simply set  $n_{cell} = 1$ . We then estimate  $n_{full}$  and  $n_{partial}$  as the median fraction of mRNA from virus among cells infected with the full and partial complement of viral genes. The rationale here is that most cells are infected with just a single virion, and so the median cell with a full RNP complement got just one full-RNP virion, and the median cell with a partial RNP complement got just one partial-RNP virion. This justifies setting these numbers as the median fractions of mRNA from virus (assuming  $n_{cell} \gg n_{full}, n_{partial}$  so that we ignore the inflation in total mRNA due to viral infection); it would not be justified to set them as the means as the mean would be influenced by co-infections. The estimates differ across timepoints (as expected since more viral mRNA will be made at later times).

```
In [2]: samples = ['10hr', '8hr', '8hr-2']

params = {
    'v_mean': 0.2,
    'n_cell': 1.0,
    '10hr': {
        'x_full': 0.67,
        'n_full': 0.11,
        'n_partial': 0.0051,
    },
    '8hr': {
        'x_full': 0.43,
        'n_full': 0.016,
        'n_partial': 0.0014,
    },
    '8hr-2': {
        'x_full': 0.42,
        'n_full': 0.042,
        'n_partial': 0.0011,
    }
}
```

### The expected distributions for each sample under the simple model

Now we plot the expected distribution of viral mRNA per cell under the simple model. Rather than trying to solve the model exactly, we just enumerate over all combinations of possible infections allowing up to 20 virions per cell (which is far more than expected to occur at any non-infectious frequency with a Poisson infection process and a mean of  $<1$ ).

We calculate the [Gini coefficient](https://en.wikipedia.org/wiki/Gini_coefficient) ([https://en.wikipedia.org/wiki/Gini\\_coefficient](https://en.wikipedia.org/wiki/Gini_coefficient)) using the [IneqPy](https://pypi.python.org/pypi/ineqPy/0.0.2) (<https://pypi.python.org/pypi/ineqPy/0.0.2>) package.

```

In [3]: import itertools
import numpy
from ineqpy import gini
%matplotlib inline
import seaborn
import matplotlib.pyplot as plt

vmax = 20

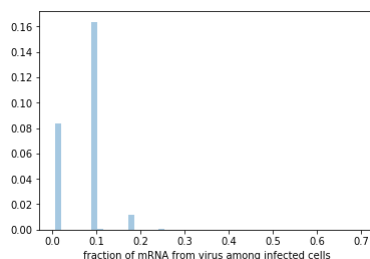
for s in samples:
    print("\nPerforming analysis for sample {}".format(s))
    weights = []
    viralfracs = []
    for (v_full, v_partial) in itertools.permutations(range(vmax + 1), 2):
        if v_full == v_partial == 0:
            continue
        weights.append(Pr_virions(params['v_mean'],
                                   params[s]['x_full'],
                                   v_full,
                                   v_partial))
        viralfracs.append(f_viral(params['n_cell'],
                                   params[s]['n_full'],
                                   params[s]['n_partial'],
                                   v_full,
                                   v_partial))
    weights = numpy.array(weights)
    viralfracs = numpy.array(viralfracs)

    ginicoeff = gini(income=viralfracs, weights=weights)
    print("The Gini coefficient is {0:.2f}".format(ginicoeff))

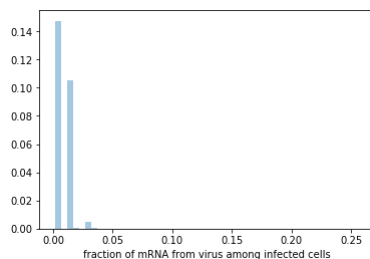
    print("Here is a plot of the distribution:")
    seaborn.distplot(viralfracs, kde=False, bins=50, hist_kws={'weights':weights})
    plt.xlabel('fraction of mRNA from virus among infected cells')
    plt.show()

```

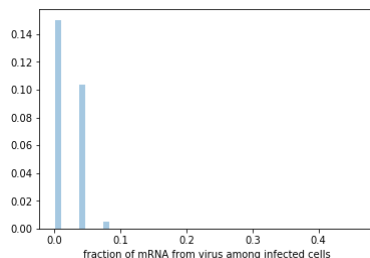
Performing analysis for sample 10hr  
The Gini coefficient is 0.77  
Here is a plot of the distribution:



Performing analysis for sample 8hr  
The Gini coefficient is 0.75  
Here is a plot of the distribution:



Performing analysis for sample 8hr-2  
The Gini coefficient is 0.87  
Here is a plot of the distribution:



## Conclusions

The simple model can generate highly heterogeneous distributions, with Gini coefficients as large or larger than those observed in the real data. However, the simple model still does not capture the continuous skew observed in the real data. Under the simple model there are two main peaks: cells infected with a virion lacking RNP, and cells infected with a virion with full RNP. There are then a few more minor peaks for double infections, and almost nothing beyond that since the MOI is so low.

In contrast, in the real data there is very broad spread, indicating additional processes layered on top of these sources of variability. Specifically, there must be additional viral, host, or stochastic factors that also spread out the amount of viral mRNA produced even in cells that do or do not receive the full RNP complement of genes.

In [ ]: