Simulations of antibiotic resistance evolution

Steven Court

March 31, 2016

Abstract

In the last 2 weeks I have been learning to use and commenting Bartek's code, and have written some scripts to analyse the data it outputs (expected number of mutations per well and speed of front). Here are some thoughts and results so far. It's currently written mainly for myself so sorry if it's a bit messy still.

1 The program

The program aims to simulate the evolutionary experiments being carried out by Bartek. In it, a chain of 24 wells are connected by small channels, with an initial inoculation of bacteria placed in the first well. An exponentially increasing concentration profile of antibiotic is imposed across the wells. Each well is given an initial amount of food, with 1 unit of food being consumed per replication event, and food is not allowed to diffuse to neighbouring wells. Food only drives replication and is not necessary for motility – bacteria are free to move randomly at a rate fixed by a diffusion constant and are point-like, non-interacting particles. Motility is also not affected by the antibiotic or any sort of chemotaxis. Mutations are coupled to birth events, and are stored as a list of mutated positions on the genome. Currently, death is linked only to birth events and depends on the presence of antibitotics.
The basic structure of the program consists of Wells, Genotypes and Cells. Cells are simulated individually and each has a well, position and genotype associated to it. Each Well has variables for antibiotic concentration and amount of food present. A Genotype contains a sequence (list of mutated genomic positions), the genotype from which it mutated, number of driver mutations, population size and 4 methods governing birth, death and mutation (see below).

An overview of the model parameters is given in Table 1. The antibiotic concentration is constructed with an exponential profile as in the experiments (though this is not easily verifiable), and takes the value in well i:

$$\text{well[i].a} = \text{uni\_antibiotic} + \text{max\_antibiotic}^{[\text{i}/\text{N}_{\text{wells}}]}, \tag{1}$$

where N_{wells} is the number of wells.

1.1 Overview of methods

It took me a while to understand exactly how gowth and death was being implemented so I'm writing this here mainly for myself. Each genotype has the following 3 methods associated with it. All can vary across the wells but none do so currently:

growth(int well): Currently returns max_growth_rate, which is 2, for all genotypes.

death(int well): Currently set to zero for all genotypes in all wells.

death_during_rep(int well): Currently death is only implemented upon reproduction and is only non-zero in the presence of antibiotics. For wildtype, MIC=20 ng ml^{-1} while for any genotype with 1 or more driver mutations, MIC=100 ng ml^{-1}. The death rate is then calculated for each well, i:

$$d[i] = \left(\frac{\text{well[i].antibiotic}}{\text{MIC}}\right)^2, \tag{2}$$

with the condition that $d \leq 1$.

In the main procedure, **main_proc()**, the actual birth rate of a cell is given by **growth()**×**growth_function(well[i].food / food$_0$)**, where the function **growth_function()** currently only returns the parameter it is given. Thus the birth rate is currently: 2×(food/food$_0$). The death rate is simply from **death()**, and is thus zero for all genotypes.

Parameter	Value	Notes
Genome length, L	10000	Needs to be >> than expected number of mutations per cell
max_no_drivers	10	Maximum number of driver mutations
Mutation probability, γ	1e-2	In the absence of Cipro this has been measured as 2e-3 and Bartek has measured values for a variety of Cipro concentrations
Diffusion, D	1.5 $\text{mm}^2\ \text{h}^{-1}$	E. coli diffusion constant in what medium?
max_growth_rate	2 h^{-1}	
Initial innoculation, n_i	100	
food_0	1e5	Available food per well (no diffusion between wells and is not replenished)
uni_ and max_antibiotic	0 and 50, respectively	Control antibiotic concentration profile, Eq. (1)

Table 1: **List of parameters and their *base values*.** Fixed from the experiment are the number of wells, 24; their dimensions, $3.5 \times 3.5 \times 10$ mm; and the dimensions of the connecting chanels, $1 \times 1 \times 1$ mm, located at the top of the wells.

1.2 Possible bugs / issues

(1) Genotype() method for mutation upon replication. Method was not always implementing the correct number of mutations per birth event. This would only have been important for smaller genome sizes, L, and fixing this did not affect the plateau observed in the expected number of mutations per well.

(2) Multiple entries stored for identical Genotypes (see genotypes_x_x.dat files). Doesn't affect results when looking at number of mutations though. Can write a script to fix this after the simulation in any case.

(3) Diffusion happens in each coordinate with a random magnitude chosen from gauss() method. If the diffusion parameter is set too high (as when I was testing between the timescales t_f and t_c) this can place bacteria far outside of the defined wells.

(4) Are we correctly using the Gillespie algorithm? At each time step, only 1 cell is chosen. This cell always moves a random amount (calculated from gauss() method), but only birth and death make up the total rate that is used to sample a random time interval, dt, from a Poisson distribution. Should *every* cell not be allowed to move in this time interval? (With only the chosen cell being allowed to replicate, die, mutate...). Also, is the total rate we are using, max_death_rate+max_growth_rate, correct? The max death rate was just arbitrarily set to be equal to the max growth rate, 2, when in fact the methods above do not allow it to exceed 1. I edited the code to update the position of every cell, which now requires the gauss() method to be called $3 \times N_{\text{cells}}$ at every timestep, with $6N_{\text{cells}}$ random numbers being generated. This massively slows down the simulation. Simulating 24 wells, with food_0=1e3 for only 4 hours takes about 2.5 minutes. A single experiment for 4 days would thus take around 15 minutes to simulate, and increasing food_0 to more realistic values of 1e5-1e8 could be problematic.

(5) In addition to above, Rosalind spotted another possible issue with the way in which the time interval is determined: a random number is drawn from a Poisson distribution and then divided by N, the total number of cells. Is this equivalent to the standard Gillespie algorithm where each cell is given it's own waiting time? I can't see how it is...

(6) Rosalind also noticed that mutations aren't being implemented correctly. In the code, for each replication event there is a probability that there is a mutation event and a determined number of mutations is chosen from a Poisson distriubtion. However all of these mutations go to only one daughter cell, leaving the mother cell completely unchanged (and thus this cell is actually unable to acquire mutations). This will certainly affect the expected number of mutations per well.

(7) I must have run a couple hundred simulations successfully, but 2 of them have crashed due to a "Segmentation fault". I do not understand this.

1.3 Possible modifications / things to think about.

(1) Include degradation of antibitotics. Are antibiotics used up by the bacteria also?
(2) Antibiotic mode: kills cells or prevents division (thus could affect birth or death rates).
(3) Antibiotic concentration alters mutation probability.
(4) Could antibiotic or food nutrient concentrations affect motility?
(5) Growth rates of specific mutants have been measured - implement these.

2 Expected number of mutations per well

One outcome Bartek was interested in was the expected number of mutations per well and how this could be affected by changing the parameters of the model. Figure 1 shows an example of the plots I have produced from some simulations. Each graph has a defined cutoff value and we only count point mutations present with an abundance above this value. Each blue point corresponds to the number of detected PMs in a specific well, at the end of a specific experiment. The red points and error bars correspond to the mean and standard deviation across all experiments.

Bartek initially expected a linear increase throughout the wells, though after testing multiple alterations to the code and thinking more carefullly about it (see notebook), I think we agree that the situation is more complicated (but also I should say that in some plots it's not obvious that there isn't a linear increase; maybe I need to collect more data per plot). There are two things to consider:

(1) The question of how many PMs will there be with an abundance above some cutoff value is not the same as the simple estimate of how many PMs we expect to see in a population after some time (which would increase linearly). Bartek has already done a calculation related to this idea with his growing tumours.

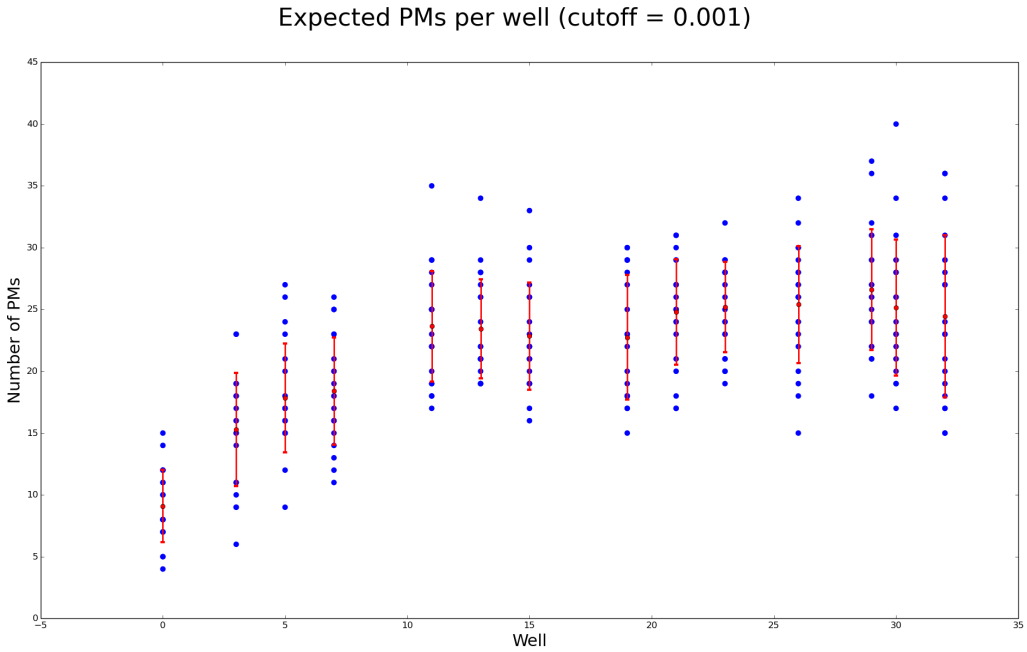
(2) There is a question of timescales, that arises from the spatial structure of the experiment. Let me define the time for a bacteria to fill a well as τ_f and the time for a bacteria to diffuse between wells as τ_d. If we are in the regime in which $\tau_d \gg \tau_f$ we would expect the entire population to be well-mixed and thus expect roughly the same number of mutations per well. On the otherhand, the situation is more complicated when $\tau_f \gg \tau_d$. Here, we expect a cell to replicate and fill the well before any cell migrates to the subsequent well. Thus mutations are allowed to accumulate as in a well mixed population, with one bacteria then being selected to seed the next population (a bit like the evolutionary experiments on plates, where some clones are re-plated every 24h). I've been thinkning how to calculate this but haven't done so yet.

3 Matching diffusion in simulation and experiment

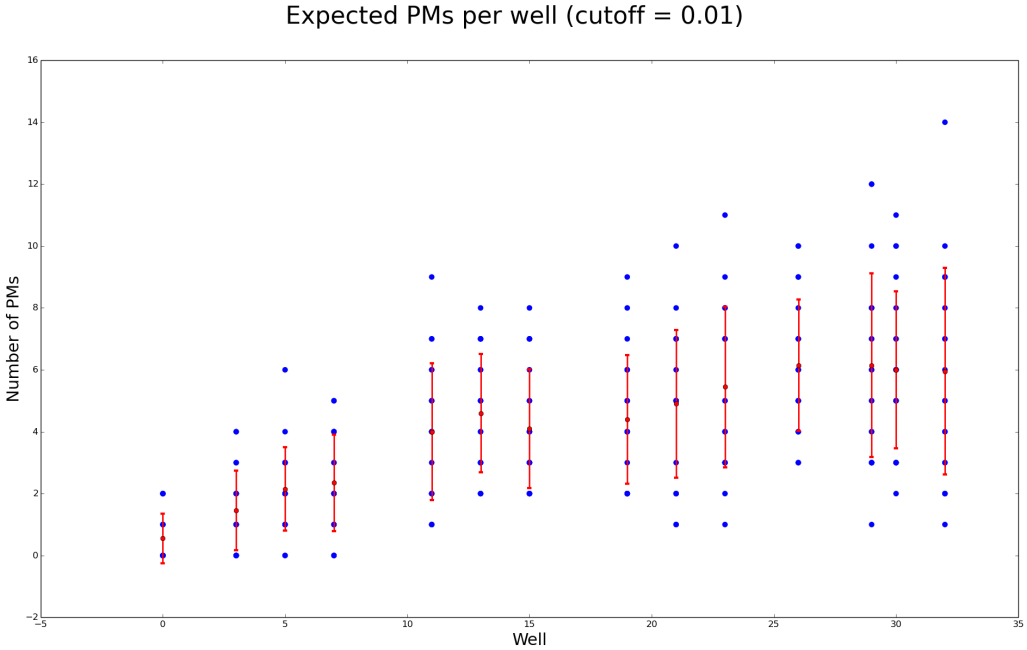
Another property Bartek wanted to check was how to go about matching the diffusion of bacteria in the simulation to the experiments. Originally I did this by printing out the population of each well n_i at regular intervals throughout the simulation. I define a well as having been colonised at the first time at which $n_i > 0$ (changing this to $n_i > 100$ doesn't change much) and store all the intervals Δt_i for each well. I have just been doing one experiment with 60 wells and average all of these intervals to find the average colonisation time between consecutive wells. I calculate the speed of the population front as 1 over this value.

Doing this for the values of bacterial diffusion (1.5, 3.0, 6.0) produces speeds of about (0.52, 0.63 and 0.75) wells h^{-1} which are all much smaller than the value of 1.5 wells h^{-1} Bartek found in (only one) of his experiments. I looked at the code again and noticed a possible bug, (4) above. Making this alteration increased v to 8.84 wells h^{-1} from 0.52 in the case where the bacterial diffusion constant was 1.5 mm^2 h^{-1}. This is now much higher than in Bartek's experiment.

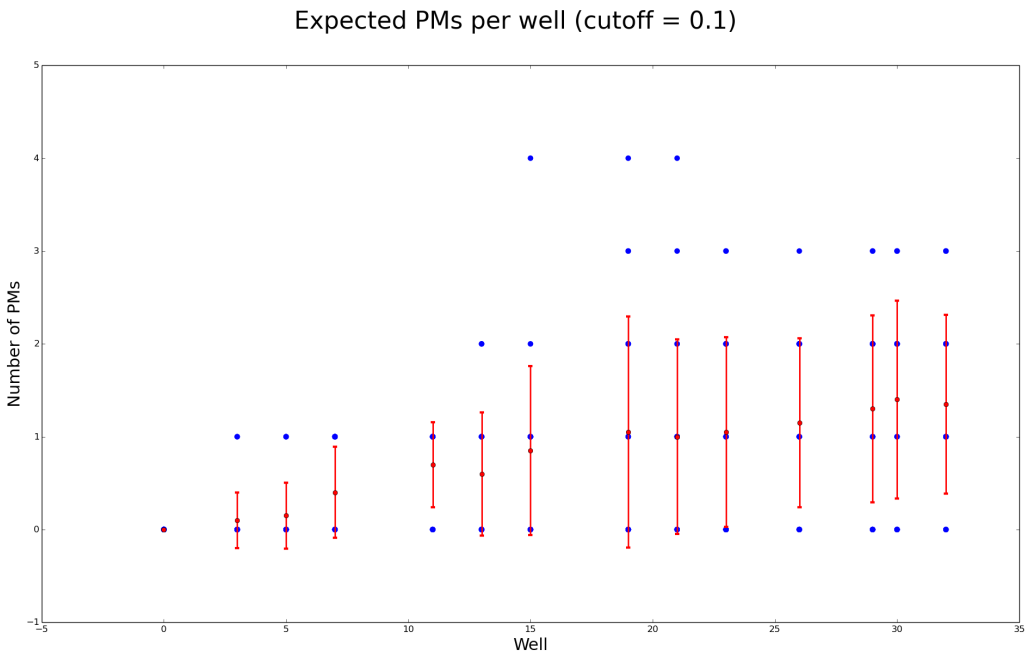
Now I have written a script to plot the wavefront at various times throughout the simulation as well as a plotting the time at which each well exceeds some critical population number, n^*. The gradient of this plot gives the speed of the wave and agrees with what I was doing above. Figure 2 shows these plots for a simulation with 100 wells, a diffusion constant of 1.5, $food_0$=1e3 and no antibiotics.


Expected PMs per well (cutoff = 0.001)
Number of PMs
Well

(a) 1a

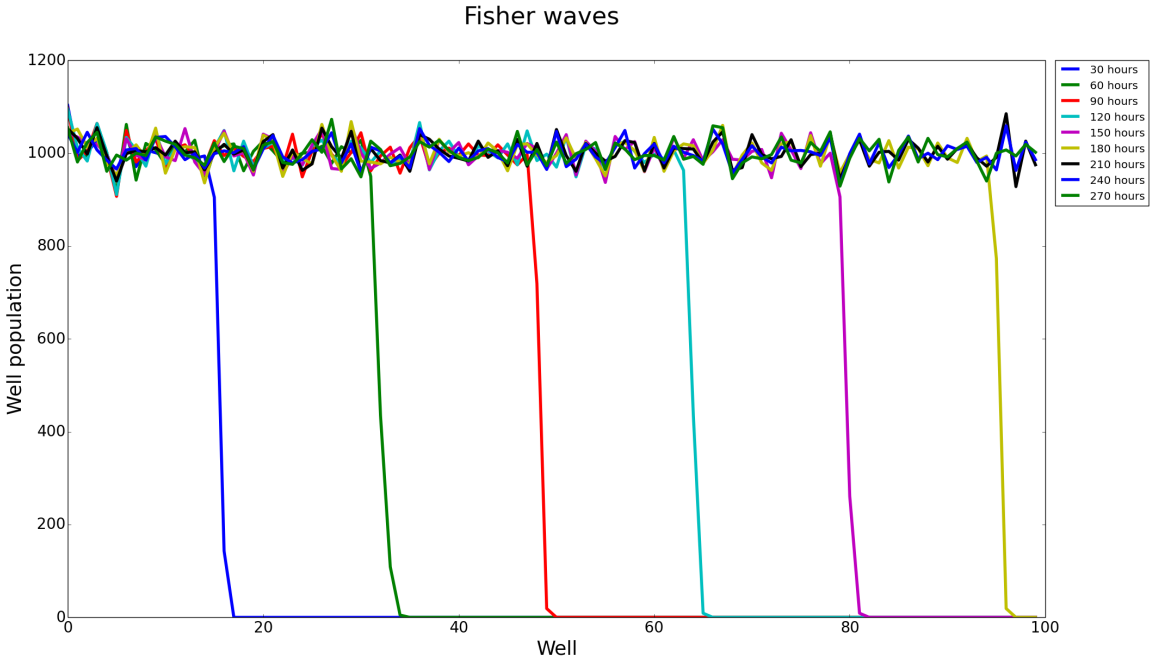

Expected PMs per well (cutoff = 0.01)
Number of PMs
Well

(b) 1b

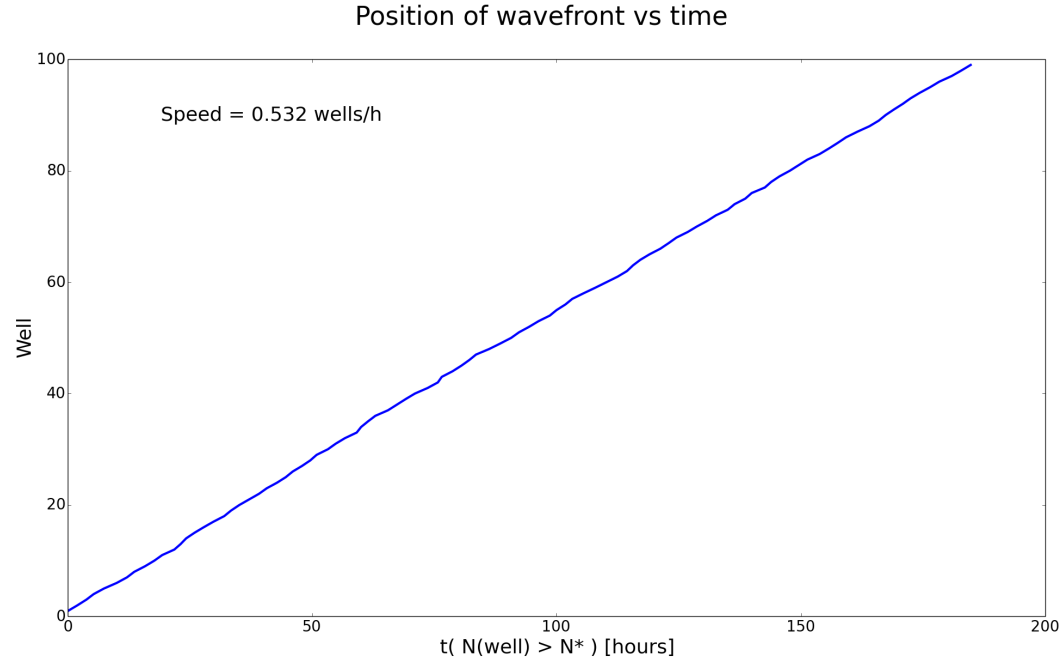


(c) 1c

Figure 1: Expected number of PMs, per well, for various cutoff abundances. (a) Shows an apparent plateau, but (b) and (c) show how this is not as obvious, or is not present, for higher cutoff values. Simulation was extended to 33 wells, with no antibiotic, for a duration of 3 days. All other parameters as in "base set".



(a) 2a



(b) 2b

Figure 2: (a) Well populations at various times during experiment. Wavefront is very steep. (b) Time at which critical population ($n^* = 100$) reaches specific wells. Gradient gives speed of wave as 0.53 wells h^{-1} when bacteria diffusion constant is 1.5 (though note that plot is produced using *incorrect* code in which position of only one bacteria is updated each timestep).