Solutions for Problem Set 2

Written by: Anita Kulkarni and Benjamin Good
(last updated on February 12, 2021)

Sample code is provided at the end of the document.

Problem 1: Measuring the per-base-pair mutation rate with the Luria Delbrück fluctuation test

Part (a)

Answers can vary, of course, but here are a few examples:

Plate ID	Number of Resistant Colonies
M6	200
M47	266
O56	175
P37	100
P63	157
Q4	127
Q29	200
S34	500

Part (b)

If we observe zero resistant colonies in a particular population, then we know that there could not have been any mutations generated in any of the cells at any point during the grow-up phase. In particular, recall from problem set 1 that $M_T = 2^T \sum_{t=1}^{T} m(t)2^{-t}$; notice that $M_T = 0$ only when $m(t) = 0$ for all t. Since $m(t)$ is Poisson distributed,

$$P(m(t) = 0) = e^{-\theta(t)} = \exp\big(-U_{\Delta\text{URA3}} N_0 2^t\big)$$

$$\implies P(M_T = 0) = \prod_{t=1}^{T} P(m(t) = 0) = \prod_{t=1}^{T} \exp\big(-U_{\Delta\text{URA3}} N_0 2^t\big) = \exp\left(-U_{\Delta\text{URA3}} N_0 \sum_{t=1}^{T} 2^t\right)$$

$$= \exp\left(-U_{\Delta\text{URA3}} N_0 \frac{2(1-2^T)}{1-2}\right) = \exp\big(-2U_{\Delta\text{URA3}} N_0 (2^T - 1)\big) = p_0$$

Recall that when jackpots are rare ($N_0 U_{\Delta\text{URA3}} \ll 1$), they will not typically occur in an experiment, while they are included in the theoretical $\langle M_T \rangle$; hence, the coefficient of variation of M_T diverges as $N_0 U_{\Delta\text{URA3}} \to 0$. On the other hand, when jackpots are rare, we would expect to see finite, nonzero values ($\to 1$) for both $\overline{p}_0$ and p_0, and these quantities are less sensitive to the number and size of jackpots than $\overline{M}_T$ is. Thus, the coefficient of variation of $\overline{p}_0$ would be finite (specifically, it would approach 0) in the limit of rare jackpots, making $\overline{p}_0$ a more robust quantity to measure in this limit than $\overline{M}_T$.

Part (c)

If we rearrange the answer from part B, we obtain

$$\log p_0 = -2U_{\Delta\text{URA3}} N_0(2^T - 1) \implies U_{\Delta\text{URA3}} = -\frac{\log p_0}{2N_0(2^T - 1)}$$

By substituting the observed value of $\overline{p}_0$ for p_0, we obtain a plug-in estimator for $U_{\Delta\text{URA3}}$:

$$\hat{U}_{\Delta\text{URA3}} = -\frac{\log \overline{p}_0}{2N_0(2^T - 1)}$$

To find the mean and variance of $\hat{U}_{\Delta\text{URA3}}$, we note that $\overline{p}_0$ is binomially distributed with mean p_0 and variance $p_0(1-p_0)/n$. In the limit that $np_0 \gg 1$ (i.e. we observe more than a few plates with zero counts), the central limit theorem implies that

$$\overline{p}_0 \approx p_0 + \underbrace{\sqrt{\frac{p_0(1-p_0)}{n}} Z}_{\delta\overline{p}_0} \tag{1}$$

where Z is a standard gaussian random variable and $\delta\overline{p}_0 \ll p_0$. This implies that we can Taylor expand the logarithm in the formula for $\hat{U}$ to obtain

$$\begin{aligned}
\hat{U}_{\Delta\text{URA3}} &= -\frac{\log(\overline{p}_0)}{2N_0(2^T-1)} = -\frac{\log(p_0 + \delta\overline{p}_0)}{2N_0(2^T-1)} && (2)\\
&= -\frac{\log(p_0)}{2N_0(2^T-1)} - \frac{\log\left(1 + \frac{\delta\overline{p}_0}{p_0}\right)}{2N_0(2^T-1)} && (3)\\
&\approx -\frac{\log(p_0)}{2N_0(2^T-1)} - \frac{\delta\overline{p}_0}{p_0 \cdot 2N_0(2^T-1)} && (4)\\
&\approx -\frac{\log(p_0)}{2N_0(2^T-1)} - \sqrt{\frac{1-p_0}{np_0\left(2N_0(2^T-1)\right)^2}} \cdot Z && (5)
\end{aligned}$$

which shows that

$$\begin{aligned}
\langle \hat{U}_{\Delta\text{URA3}} \rangle &\approx U_{\Delta\text{URA3}} && (6)\\
\text{Var}(\hat{U}_{\Delta\text{URA3}}) &\approx \frac{1-p_0}{np_0\left(2N_0(2^T-1)\right)^2} = \frac{1-p_0}{np_0 \log^2\left(\frac{1}{p_0}\right)} \cdot U^2_{\Delta\text{URA3}} && (7)
\end{aligned}$$

or, for the coefficient of variation,

$$c_V = \frac{\sqrt{\text{Var}(\hat{U}_{\Delta\text{URA3}})}}{\langle \hat{U}_{\Delta\text{URA3}} \rangle} \approx \sqrt{\frac{1-p_0}{np_0 \log^2\left(\frac{1}{p_0}\right)}} \tag{8}$$

Of the $n = 720$ plates in this experiment, a fraction $\overline{p}_0 \approx 0.37$ had zero colonies. Plugging this into our estimate for $\hat{U}_{\Delta\text{URA3}}$ and $\hat{c}_V$, we obtain

$$\hat{U}_{\Delta\text{URA3}} \approx 3 \times 10^{-8} \tag{9}$$

$$\hat{c}_V \approx 0.05 \tag{10}$$

As expected, the estimated coefficient of variation is much less than 1, justifying our use of the $\delta\overline{p}_0/p_0$ expansion above.

Part (d)

Note that the total SNV rate in the gene $\times$ fraction of SNVs that are nonsense $\equiv$ observed nonsense rate. If we let μ be the per-base-pair SNV rate, then the total SNV rate (in the gene) is 804 bp $\times\mu$. The fraction of nonsense SNVs is $\frac{123}{2412}$, making the left hand side $\frac{123}{2412} \times 804\mu = 41\mu$. The right hand side is $\hat{U}_{\Delta\text{URA3}} \times \frac{64}{237}$. Plugging in our estimate for $\hat{U}_{\Delta\text{URA3}}$ from the previous part, we get

$$\mu \approx 2.01 \times 10^{-10} \frac{\text{mutations}}{\text{site} \times \text{generation}}$$

Part (e)

Of the 237 loss-of-function mutations, 167 were caused by SNVs. Thus, the rate of loss-of-functions caused by SNVs per gene per generation $= \hat{U}_{\Delta\text{URA3}} \times \frac{167}{237}$. Now we use a similar line of logic as in the previous part. The total mutation rate (over the gene) $\times$ the fraction of protein function disrupting SNVs (i.e. probability that a random SNV disrupts protein function) $= \hat{U}_{\Delta\text{URA3}} \times \frac{167}{237}$. Our quantity of interest is

$$\hat{U}_{\Delta\text{URA3}} \times \frac{167}{237} \times \frac{1}{804\mu} \approx 0.133$$

Problem 2: Universality and non-universality among serial dilution models

Part (a)

The wildtype individuals each have fitness $\sim$ Gaussian(r, σ^2); call this F_0. The mutant individuals each have fitness $\sim$ Gaussian$(r + s, \sigma^2 + \nu)$; call this F_1. Denote N_0 and N_1 as the total numbers of wildtype and mutant individuals, respectively, where $N_0 + N_1 = N$, and $f = N_1/N$. At the end of a grow-up phase, just before dilution, the total mutant frequency is:

$$f(\Delta t) = \frac{N_1(\Delta t)}{N_0(\Delta t) + N_1(\Delta t)} = \frac{\sum_i^{N_1} e^{F_{1,i}\Delta t}}{\sum_j^{N_0} e^{F_{0,j}\Delta t} + \sum_i^{N_1} e^{F_{1,i}\Delta t}} = \frac{\frac{1}{N}\sum_i^{N_1} e^{F_{1,i}\Delta t}}{\frac{1}{N}\sum_j^{N_0} e^{F_{0,j}\Delta t} + \frac{1}{N}\sum_i^{N_1} e^{F_{1,i}\Delta t}}$$

Examine each sum individually:

$$\frac{1}{N}\sum_i^{N_1} e^{F_{1,i}\Delta t} = f\frac{1}{Nf}\sum_i^{Nf} e^{F_{1,i}\Delta t} = f\frac{1}{Nf}\sum_i^{Nf} \exp\left[\left(r + s + \sqrt{\sigma^2 + \nu}Z_{1,i}\right)\Delta t\right]$$

$$= fe^{(r+s)\Delta t}\frac{1}{Nf}\sum_{i}^{Nf}\exp\left[\Delta t\sqrt{\sigma^2+\nu}Z_{1,i}\right]$$

The argument of $\exp\left[\Delta t\sqrt{\sigma^2+\nu}Z_{1,i}\right]$ is not necessarily small, so it is most rigorous to not expand the exponential and to treat the whole expression as a random variable X_i. The sum then becomes

$$fe^{(r+s)\Delta t}\frac{1}{Nf}\sum_{i}^{Nf}X_i$$

but since $Nf \gg 1$, we can use the central limit theorem on $\frac{1}{Nf}\sum_i^{Nf}X_i$ so that

$$fe^{(r+s)\Delta t}\frac{1}{Nf}\sum_{i}^{Nf}X_i \approx fe^{(r+s)\Delta t}\left[\langle X_i\rangle + \sqrt{\frac{\mathrm{Var}(X_i)}{Nf}}Z_1\right]$$

The moments of X_i can be calculated by hand or by recognizing that X_i is a log-normal random variable and utilizing Wikipedia:

$$\langle X_i\rangle = e^{\frac{1}{2}(\sigma^2+\nu)(\Delta t)^2}$$

$$\mathrm{Var}(X_i) = e^{(\sigma^2+\nu)(\Delta t)^2}\left[e^{(\sigma^2+\nu)(\Delta t)^2}-1\right]$$

Plug these back into our expression:

$$fe^{(r+s)\Delta t}\left[\langle X_i\rangle + \sqrt{\frac{\mathrm{Var}(X_i)}{Nf}}Z_1\right] = fe^{(r+s)\Delta t+\frac{1}{2}(\sigma^2+\nu)(\Delta t)^2}\left[1+\sqrt{\frac{e^{(\sigma^2+\nu)(\Delta t)^2}-1}{Nf}}Z_1\right]$$

If we follow an analogous set of steps for $\sum_j^{N_0}e^{F_{0,j}\Delta t}$, we get that

$$\sum_{j}^{N_0}e^{F_{0,j}\Delta t} \approx (1-f)e^{r\Delta t+\frac{1}{2}\sigma^2(\Delta t)^2}\left[1+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N(1-f)}}Z_0\right]$$

Plug these expressions back into the original expression for $f(\Delta t)$:

$$f(\Delta t) \approx \frac{fe^{(r+s)\Delta t+\frac{1}{2}(\sigma^2+\nu)(\Delta t)^2}\left[1+\sqrt{\frac{e^{(\sigma^2+\nu)(\Delta t)^2}-1}{Nf}}Z_1\right]}{(1-f)e^{r\Delta t+\frac{1}{2}\sigma^2(\Delta t)^2}\left[1+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N(1-f)}}Z_0\right]+fe^{(r+s)\Delta t+\frac{1}{2}(\sigma^2+\nu)(\Delta t)^2}\left[1+\sqrt{\frac{e^{(\sigma^2+\nu)(\Delta t)^2}-1}{Nf}}Z_1\right]}$$

$e^{r\Delta t+\frac{1}{2}\sigma^2(\Delta t)^2}$ can be canceled from both the numerator and denominator, and we can cancel ν inside the square roots (this will make it easier to combine terms later) since $\nu \ll \sigma^2$:

$$\approx \frac{fe^{s\Delta t+\frac{1}{2}\nu(\Delta t)^2}\left[1+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1\right]}{(1-f)\left[1+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N(1-f)}}Z_0\right]+fe^{s\Delta t+\frac{1}{2}\nu(\Delta t)^2}\left[1+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1\right]}$$

Begin expanding $f(\Delta t)$ to leading order in s, ν, and $1/N$:

$$\approx \frac{f\left[1+s\Delta t+\frac{1}{2}\nu(\Delta t)^2\right]\left[1+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1\right]}{(1-f)\left[1+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N(1-f)}}Z_0\right]+f\left[1+s\Delta t+\frac{1}{2}\nu(\Delta t)^2\right]\left[1+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1\right]}$$

$$= \frac{f+fs\Delta t+\frac{1}{2}f\nu(\Delta t)^2+f\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1+\text{h.o.t.}}{1-f+(1-f)\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N(1-f)}}Z_0+f+fs\Delta t+\frac{1}{2}f\nu(\Delta t)^2+f\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1+\text{h.o.t.}}$$

Expand as $1/(1+x)\approx 1-x$:

$$f(\Delta t)\approx\left[f+fs\Delta t+\frac{1}{2}f\nu(\Delta t)^2+f\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1\right]$$

$$\times\left[1-fs\Delta t-\frac{1}{2}f\nu(\Delta t)^2-f\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1-(1-f)\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N(1-f)}}Z_0\right]$$

$$\approx f+fs\Delta t+\frac{1}{2}f\nu(\Delta t)^2+f\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1-f^2s\Delta t$$

$$-\frac{1}{2}f^2\nu(\Delta t)^2-f^2\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1-f(1-f)\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N(1-f)}}Z_0$$

$$= f+f(1-f)s\Delta t+\frac{1}{2}f(1-f)\nu(\Delta t)^2+f(1-f)\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{Nf}}Z_1-f(1-f)\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N(1-f)}}Z_0$$

$$= f+f(1-f)\Delta t\left(s+\frac{1}{2}\nu\Delta t\right)+f(1-f)\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N}}\left(\frac{1}{\sqrt{f}}Z_1-\frac{1}{\sqrt{1-f}}Z_0\right)$$

$$= f+f(1-f)\Delta t\left(s+\frac{1}{2}\nu\Delta t\right)+\sqrt{\frac{e^{\sigma^2(\Delta t)^2}-1}{N}f(1-f)}Z$$

We're not done yet - there is still the dilution step remaining! And in any case, notice that when $\sigma\to 0$, the noise term cancels out, which is not what we want.

$$f(k+1)=\frac{\text{Poisson}(Nf(\Delta t))}{\text{Poisson}(N[1-f(\Delta t)])+\text{Poisson}(Nf(\Delta t))}$$

We can use the Gaussian approximation to the Poisson distribution since the λ parameter is large in each case:

$$\approx\frac{Nf(\Delta t)+\sqrt{Nf(\Delta t)}Z_1'}{N[1-f(\Delta t)]+\sqrt{N[1-f(\Delta t)]}Z_0'+Nf(\Delta t)+\sqrt{Nf(\Delta t)}Z_1'}$$

$$= \frac{f(\Delta t) + \sqrt{\frac{f(\Delta t)}{N}} Z_1'}{1 - f(\Delta t) + \sqrt{\frac{1-f(\Delta t)}{N}} Z_0' + f(\Delta t) + \sqrt{\frac{f(\Delta t)}{N}} Z_1'}$$

Using similar methods as above ($1/(1+x) \approx 1 - x$, taking lowest-order terms in $1/N$), we find that

$$f(k+1) \approx f(\Delta t) + \sqrt{\frac{f(\Delta t)(1 - f(\Delta t))}{N}} Z'$$

Plug in $f(\Delta t)$ from above:

$$f(k+1) \approx f + f(1-f)\Delta t \left(s + \frac{1}{2}\nu \Delta t \right) + \sqrt{\frac{e^{\sigma^2 (\Delta t)^2} - 1}{N} f(1-f)} Z + \sqrt{\frac{f(1-f)}{N}} Z'$$

Notice that $\sqrt{\frac{f(1-f)}{N}} Z'$ is the lowest-order term when our expression is plugged into $\sqrt{\frac{f(\Delta t)(1-f(\Delta t))}{N}} Z'$. Combining the two Gaussian variables gives us a final answer of

$$f(k+1) \approx f + f(1-f)\Delta t \left(s + \frac{1}{2}\nu \Delta t \right) + \sqrt{\frac{e^{\sigma^2 (\Delta t)^2}}{N} f(1-f)} Z_k$$

We can see that

$$\langle f(k+1) \rangle \approx f + f(1-f)\Delta t \left(s + \frac{1}{2}\nu \Delta t \right)$$

and

$$\mathrm{Var}(f(k+1)) = \frac{e^{\sigma^2 (\Delta t)^2}}{N} f(1-f)$$

Also, it is clear that since the form (f, N dependence) of $f(k+1)$ is the same as that of the original serial dilution model, the two models lie in the same universality class. In units of cycles,

$$s_{\text{eff}} = \Delta t \left(s + \frac{1}{2}\nu \Delta t \right), N_{\text{eff}} = N e^{-\sigma^2 (\Delta t)^2}$$

In units of generations,

$$s_{\text{eff}} = s + \frac{1}{2}\nu \Delta t, N_{\text{eff}} = N \Delta t e^{-\sigma^2 (\Delta t)^2}$$

Part (b)

This problem starts out similarly part A (we are still using Fs the same way), except that since fitness perturbations are shared across all individuals in the flask, the summations over the exponents are treated differently:

$$f(\Delta t) = \frac{\sum_i^{N_1} e^{F_1 \Delta t}}{\sum_j^{N_0} e^{F_0 \Delta t} + \sum_i^{N_1} e^{F_1 \Delta t}} = \frac{N_1 e^{F_1 \Delta t}}{N_0 e^{F_0 \Delta t} + N_1 e^{F_1 \Delta t}} = \frac{f e^{(F_1 - F_0)\Delta t}}{1 - f + f e^{(F_1 - F_0)\Delta t}}$$

Since $(F_1 - F_0)\Delta t$ has a small mean and variance, we can use the $1/(1+x) \approx 1 - x$ approximation:

$$f(\Delta t) \approx f e^{(F_1 - F_0)\Delta t} (1 - f e^{(F_1 - F_0)\Delta t} + f)$$

Now let's use the fact that $F_1 - F_0 \sim \text{Gaussian}(s, \nu) = s + \sqrt{\nu}Z$ and expand $f(\Delta t)$ to leading order in s and ν, starting with the fact that $e^{(F_1-F_0)\Delta t} \approx 1 + s\Delta t + \sqrt{\nu}\Delta t Z$:

$$f(\Delta t) \approx f(1+s\Delta t+\sqrt{\nu}\Delta t Z)[1-f(1+s\Delta t+\sqrt{\nu}\Delta t Z)+f] = f(1+s\Delta t+\sqrt{\nu}\Delta t Z)(1-fs\Delta t-f\sqrt{\nu}\Delta t Z)$$

$$\approx f(1 + s\Delta t + \sqrt{\nu}\Delta t Z - fs\Delta t - f\sqrt{\nu}\Delta t Z) = f + f(1-f)s\Delta t + f(1-f)\sqrt{\nu}\Delta t Z$$

We've already done most of the work of the dilution step in part A, so simply plug in the last part:

$$f(k+1) \approx f + s\Delta t f(1-f) + f(1-f)\sqrt{\nu}\Delta t Z + \sqrt{\frac{f(1-f)}{N}}Z'$$

Combining the two Gaussian variables gives us a final answer of

$$f(k+1) \approx f + s\Delta t f(1-f) + \sqrt{\frac{f(1-f)}{N}}\sqrt{N\nu(\Delta t)^2 f(1-f) + 1}Z_k$$

Thus,

$$\langle f(k+1)\rangle = f + s\Delta t f(1-f)$$

and

$$\text{Var}(f(k+1)) = \frac{f(1-f)}{N}\left[N\nu(\Delta t)^2 f(1-f) + 1\right]$$

This model is clearly not in the same universality class as the original serial dilution model because of the extra $\sqrt{f(1-f)}$ dependence in the Z_k term.

Problem 3: Neutral mutation accumulation in individuals vs. populations

Part (a)

The probability of any individual having a mutation at site ℓ is $f_\ell(t)$. The expected number of mutations a randomly sampled individual would have is

$$M_1(t) = \sum_{\ell=1}^{L}\left[f_\ell(t) \times 1 + (1 - f_\ell(t)) \times 0\right] = \sum_{\ell=1}^{L} f_\ell(t)$$

Part (b)

This calculation can be done analogously to that of part A, but we can also notice that the probability that site ℓ contains a mutation in both of two randomly chosen individuals is the product of the probabilities of each individual independently containing a mutation at the site. Adding this up over all sites, we get that

$$M_2(t) = \sum_{\ell=1}^{L} f_\ell^2(t)$$

The above gives the expected number of shared *sites*, but if we want the expected number of shared *mutations*, multiply by 2:

$$M_2(t) = 2\sum_{\ell=1}^{L} f_\ell^2(t)$$

The results are analogous for a random sample of n individuals:

$$M_n(t) = \sum_{\ell=1}^{L} f_\ell^n(t)$$

for shared sites, and

$$M_n(t) = n\sum_{\ell=1}^{L} f_\ell^n(t)$$

for shared mutations.

Part (c)

This will be similar to what was done in class (throughout the problem, N_e will just be written as N):

$$\frac{\partial f}{\partial t} = \mu(1-f) - \nu f + \sqrt{\frac{f(1-f)}{N}}\eta(t) \implies f(t+\delta t) = f(t) + [\mu - \mu f(t)]\delta t + \sqrt{\frac{f(t)(1-f(t))\delta t}{N}} Z_t - \nu f(t)\delta t$$

$$\implies \langle f(t+\delta t)\rangle = \langle f(t)\rangle + \mu\delta t - (\mu+\nu)\langle f(t)\rangle\delta t \implies \frac{\partial\langle f\rangle}{\partial t} = \mu - (\mu+\nu)\langle f\rangle$$

After solving this equation with an initial condition of $\langle f(0)\rangle = 0$ (since there is a clonal ancestor), we get

$$\langle f(t)\rangle = \frac{\mu}{\mu+\nu}\left(1 - e^{-(\mu+\nu)t}\right)$$

and

$$M_1(t) = L\frac{\mu}{\mu+\nu}\left(1 - e^{-(\mu+\nu)t}\right)$$

Part (d)

Start with the same SDE as before:

$$f(t+\delta t) = f(t) + \mu\delta t - (\mu+\nu)f(t)\delta t + \sqrt{\frac{f(t)(1-f(t))\delta t}{N}} Z_t$$

Square both sides and remove all terms that average to 0 (proportional to Z_t) or that are not lowest order in δt:

$$f^2(t+\delta t) = f^2(t) + 2\mu f(t)\delta t - 2(\mu+\nu)f^2(t)\delta t + \frac{1}{N}f(t)Z_t^2\delta t - \frac{1}{N}f^2(t)Z_t^2\delta t$$

and hence:

$$\begin{aligned}\frac{\partial\langle f^2(t)\rangle}{\partial t} &= 2\mu\langle f(t)\rangle - 2(\mu+\nu)\langle f^2(t)\rangle + \frac{1}{N}\langle f(t)\rangle - \frac{1}{N}\langle f^2(t)\rangle \\ &= \left(2\mu + \frac{1}{N}\right)\langle f(t)\rangle - \left[2(\mu+\nu) + \frac{1}{N}\right]\langle f^2(t)\rangle\end{aligned}$$

Plugging in for $\langle f(t)\rangle$, we have

$$\frac{\partial\langle f^2(t)\rangle}{\partial t} = \frac{\mu}{\mu+\nu}\left(2\mu + \frac{1}{N}\right)\left(1 - e^{-(\mu+\nu)t}\right) - \left[2(\mu+\nu) + \frac{1}{N}\right]\langle f^2(t)\rangle$$

This differential equation can be solved using the integrating factor method, which (for $\langle f^2(0)\rangle = 0$) yields

$$\begin{aligned}\langle f^2(t)\rangle &= \frac{\mu}{\mu+\nu}\left(2\mu + \frac{1}{N}\right)e^{-\left[2(\mu+\nu)+\frac{1}{N}\right]t}\int_0^t \left[e^{\left[2(\mu+\nu)+\frac{1}{N}\right]t'} - e^{\left(\mu+\nu+\frac{1}{N}\right)t'}\right]dt' \\ &= \frac{\mu}{\mu+\nu}\left(2\mu + \frac{1}{N}\right)\left[\frac{1 - e^{-\left[2(\mu+\nu)+\frac{1}{N}\right]t}}{2(\mu+\nu)+\frac{1}{N}} - \frac{e^{-(\mu+\nu)t} - e^{-\left[2(\mu+\nu)+\frac{1}{N}\right]t}}{\mu+\nu+\frac{1}{N}}\right]\end{aligned}$$

and hence

$$\begin{aligned}M_2(t) &= \frac{L\mu}{\mu+\nu}\left(2\mu + \frac{1}{N}\right)\left[\frac{1 - e^{-\left[2(\mu+\nu)+\frac{1}{N}\right]t}}{2(\mu+\nu)+\frac{1}{N}} - \frac{e^{-(\mu+\nu)t} - e^{-\left[2(\mu+\nu)+\frac{1}{N}\right]t}}{\mu+\nu+\frac{1}{N}}\right] \\ &= L\mu\left(\frac{2N\mu+1}{N\mu+N\nu+1}\right)\left[\frac{1 - e^{-(\mu+\nu)t}}{\mu+\nu} - \frac{1 - e^{-\left[2(\mu+\nu)+\frac{1}{N}\right]t}}{2(\mu+\nu)+\frac{1}{N}}\right]\end{aligned}$$

In laboratory settings, the parameters L, μ, and N will often be such that

$$\frac{1}{L\mu} \ll N \ll \frac{1}{\mu} \tag{11}$$

For example, for the *E. coli* populations in Problem 4 of Problem Set 1, we have something like $L \sim 10^6$, $\mu \sim 10^{-10}$, and $N \sim 10^7$. In the empirically relevant limit where $t \ll 1/\mu$, our expressions for $M_1(t)$ and $M_2(t)$ reduce to

$$M_1(t) \approx L\mu t \tag{12}$$

$$M_2(t) \approx L\mu t \cdot \left(1 - \frac{1 - e^{-t/N}}{t/N}\right) \approx \begin{cases} \frac{L\mu t^2}{2N} & \text{if } t \ll N \\ L\mu t & \text{if } t \gg N \end{cases} \tag{13}$$

This shows that the number of mutations per clone will increase at rate $L\mu$, but the number of mutations shared by a pair of clones will be much less than this until $t \gtrsim N$. The reason for this is that mutations are unlikely to be shared by a pair of random individuals unless they drift to intermediate frequencies — this requires a time of order $t{\sim}N$.

Problem 4:

Part (a)

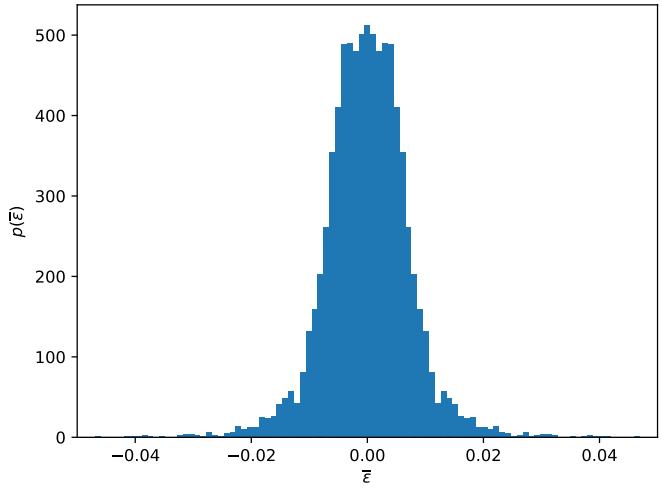
From the definitions of $\overline{\epsilon}_i$ and Δ_i, we have

$$\Delta_i \equiv \hat{s}_{i,1} - \hat{s}_{i,2} = (s_i + \epsilon_{i,1}) - (s_i + \epsilon_{i,2}) = \epsilon_{i,1} - \epsilon_{i,2} \tag{14}$$

and

$$\overline{\epsilon_i} \equiv \overline{s}_i - s_i = \frac{\hat{s}_{i,1} + \hat{s}_{i,2}}{2} - s_i = \frac{(s_i + \epsilon_{i,1}) + (s_i + \epsilon_{i,2})}{2} - s_i = \frac{\epsilon_{i,1} + \epsilon_{i,2}}{2} \tag{15}$$

If the distsribution of $\epsilon_{i,j}$ is symmetric, then $-\epsilon_{i,j}$ has the same distribution as ϵ_{ij}. This implies that $\epsilon_{i,1} - \epsilon_{i,2}$ has the same distribution as $\epsilon_{i,1} + \epsilon_{i,2}$, which implies that $\overline{\epsilon}_i$ has the same distribution as $\Delta_i/2$. By pooling our observations of Δ_i across different gene deletions, we can estimate the empirical distribution of ϵ_i:

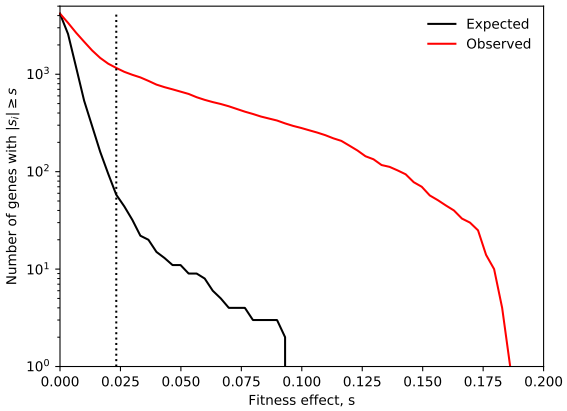


This shows that the typical errors in the fitness estimates are on the order of $\langle|\overline{\epsilon_i}|\rangle \approx 0.5\%$.

(Note: in estimating this distribution, we haved removed 255 genes in which $s_{i,1}$ or $s_{i,2}$ is less than -0.2, as we noticed that these have $\overline{\epsilon}_i$ values much higher than the other genes, although they are still small compared to $\overline{s}_i$.)

Part (b)

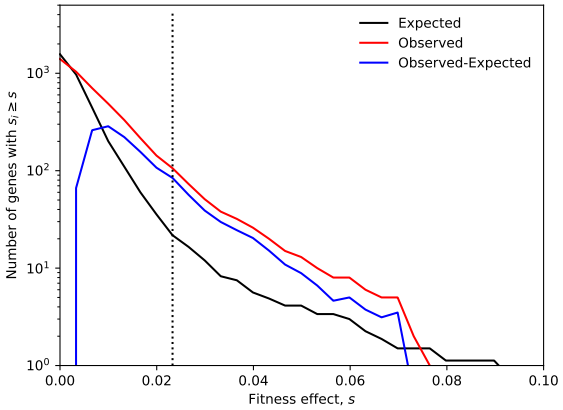
If all gene deletions were neutral ($s_i = 0$), then the distribution of $\overline{s}_i$ would be the same as the distribution of $\overline{\epsilon}_i$ that we estimated above. We can therefore use this empirical estimate to calculate the fraction of genes we would expect to observe with $|\overline{s}_i| \geq s$ under this null hypothesis, which is shown in the black line below:



By comparing to the observed distribution (red line), we see that the estimated fraction of false positives decays to 5% for fitness effects of size $s \approx 2.5\%$ (dashed line). Roughly 1100 genes ($\approx$25% of the total) have absolute fitness effects larger than this value, the vast majority of which are negative. Much larger fitness costs are also possible: on the order of a few hundred genes have fitness costs larger than 10%.

Part (c)

We can carry out the same procedure for beneficial fitness effects ($s_i \geq s$), by multiplying the expected distribution above by a factor of 0.5 (to focus on positive half of $p(\bar{\epsilon})$) and another factor of 0.75 (to account for the significantly deleterious mutations identified above). This yields:



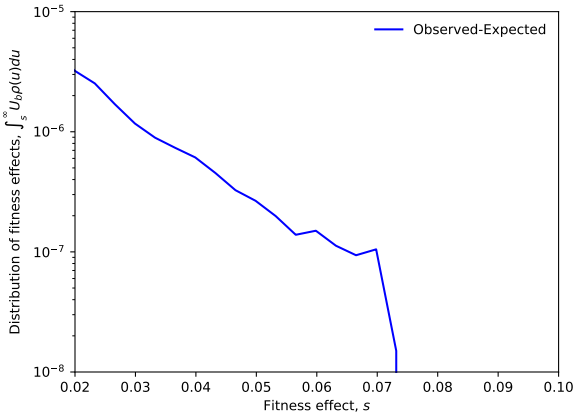
In this case, the estimated false positive rate never decays as low as 5%, so no mutations are "statistically significant" on their own. However, we still observe a roughly 5-fold enrichment of gene deletions with beneficial fitness effects larger than $\approx$2.5% (dashed line). There are about 100 observed genes here, 20 of which would be expected to arise by chance (false positive rate of 20%) leaving about 80 genes as true positives.

A rough estimate of the distribution of fitness effects of these true positives can be obtained by subtracting the expected number of false positives from the observed number of

genes with different values of $\overline{s}_i \geq s$ (blue line). This shows that the vast majority of fitness effects lie between $2.5 - 5\%$, and decay roughly exponentially with s.

Part (d)

If all beneficial mutations correspond to loss of function mutations, then the analysis in part (c) suggests that, for $s \geq 2.5\%$, we can estimate the distribution of fitness effects of beneficial gene deletions by the blue line above, but with an additional scale factor μ_Δ, which represents the mutation rate to loss of function phenotypes in a single gene. Using the estimate for the URA3 gene from Problem 1 ($\mu_\Delta \approx U_\Delta \approx 3 \times 10^{-8}$), our estimate for $U_b \rho(s)$ can be expressed as



In other words, in this enviornment, each cell has a total probability of $\sim 2 \times 10^{-6}$ of producing a beneficial mutation with $s \geq 2.5\%$.

Part (e)

If there are $\approx$5000 strains in the library, each strain should start at an average frequency of $f_0 \approx 1/5000 \approx 2 \times 10^{-4}$. A fitness effect of order $s_{\min} = \langle |\overline{\epsilon}_i| \rangle \approx 0.5\%$ would change the frequency of such a strain by

$$f(\Delta t) - f_0 = \frac{f_0 e^{s\Delta t}}{f_0 e^{s\Delta t} + (1 - f_0)} - f_0 \approx 3 \times 10^{-5}. \tag{16}$$

over the $\Delta t = 26$ generations of the assay. The analogous changes due to genetic drift are of order $\Delta f \sim \sqrt{f_0 \Delta t / N_e}$. We therefore require a population size less than $\sim \Delta t f_0 / \Delta f^2 \sim 5 \times 10^6$ to produce a frequency change comparable to the minimum resolvable fitness effect.

Similarly, we'd need a frequency resolution of at least $\Delta f \sim 3 \times 10^{-5}$ to show that the fitness effect of a truly neutral mutation is less than $s_{\min} \approx 0.5\%$.

Sample code for Problem Set 2

```
import numpy
import sys

file = open("../data_files/lang_murray_08_fluctuation_test.txt","r")
file.readline() # ignore header
colony_counts = []
for line in file:
    items = line.split(",")
    count = float(items[1])
    colony_counts.append(count)

colony_counts = numpy.array(colony_counts)

T = 13
N = 2000

n = len(colony_counts)
p0 = (colony_counts==0).sum()*1.0/n

U = -numpy.log(p0)/(2*N*(2**T))

cv = numpy.sqrt((1-p0)/(n*p0*numpy.square(numpy.log(1/p0))))

sys.stdout.write("n=%d\n" % len(colony_counts))
sys.stdout.write("pbar0 = %g\n" % p0)
sys.stdout.write("U_dURA = %g x 10^-8\n" % (U*1e08))
sys.stdout.write("cv = %g\n" % cv)
```

```
import pylab
import numpy
import sys

file = open("../data_files/qian_etal_2012_deletion_fitnesses.txt")
file.readline() # header
gene_names = []
s1s = []
s2s = []
pseudogenes = []
for line in file:
    items = line.split(",")
    gene_names.append(items[0].strip())
    s1s.append(float(items[1]))
    s2s.append(float(items[2]))

    if gene_names[0]=="*":
        pseudogenes.append(True)
    else:
        pseudogenes.append(False)

s1s = numpy.array(s1s)
s2s = numpy.array(s2s)
pseudogenes = numpy.array(pseudogenes)

bad_ss = numpy.logical_or(s1s<-0.2,s2s<-0.2)
good_idxs = numpy.logical_not(bad_ss)

print bad_ss.sum(), "strongly deleterious indices"
print good_idxs.sum(), "remaining genes"
s1s = s1s[good_idxs]
s2s = s2s[good_idxs]

# s2s-s1s = eps2 - eps1 ~ eps1+eps2 (in distribution)
# s1s+s2s/2 = sbars + (eps1+eps2)/2 in distribution.

ss = (s1s+s2s)/2
errors = numpy.fabs(s2s-s1s)
simulated_stderrs = errors/2.0

#print (ss>=0).sum(), (ss<0).sum()

sigma = (numpy.square(simulated_stderrs).mean())**0.5
standard_errors = simulated_stderrs/sigma

```

```
print "sigma =", sigma
print "<|epsbar|> =", simulated_stderrs.mean()
xs = numpy.linspace(0,40,100)
error_sf = numpy.array([(standard_errors>x).mean() for x in xs])
observed = numpy.array([(ss/sigma>x).sum() for x in xs])

observed_all = numpy.array([(numpy.fabs(ss)/sigma>x).sum() for x in xs])
expected_all = numpy.array([(standard_errors>x).sum() for x in xs])
fdr_all = (expected_all+(observed_all==0))*1.0/(observed_all+(observed_all==0))
num_deleterious = observed_all-expected_all

# Get index closest to FDR of 5%
critical_idx = numpy.fabs(fdr_all-0.05).argmin()
print "Critical s for |s_i|>s is ", xs[critical_idx]*sigma, 'FDR=%g' % fdr_all[critic
print "Number of genes is", num_deleterious[critical_idx], '(%g)' % (num_deleterious[

pylab.figure(3)
bins = numpy.linspace(-0.05,0.05,100)
pylab.hist(numpy.hstack([simulated_stderrs,-1*simulated_stderrs]),bins=bins)
pylab.xlim([-0.05,0.05])
pylab.xlabel('$\overline{\epsilon}$')
pylab.ylabel('$p(\overline{\epsilon})$')
pylab.savefig('problem_4_a.pdf')

pylab.figure(1)
pylab.semilogy(xs*sigma,expected_all,'k-',label='Expected')
pylab.semilogy(xs*sigma,observed_all,'r-',label='Observed')
pylab.semilogy([xs[critical_idx]*sigma,xs[critical_idx]*sigma],[1,observed_all[0]],'k

pylab.ylim([1,5e03])
pylab.xlim([0,0.2])
pylab.xlabel('Fitness effect, s')
pylab.ylabel('Number of genes with $|s_i| \geq s$')
pylab.legend(frameon=False,loc='upper right')
pylab.savefig('problem_4_b.pdf')

observed_ben = numpy.array([(ss/sigma>x).sum() for x in xs])
expected_ben = numpy.array([(standard_errors>x).sum()/2.0*(3.0/4) for x in xs])
fdr_ben = (expected_ben+(observed_ben==0))*1.0/(observed_ben+(observed_ben==0))

# Get index closest to FDR of 10%
critical_idx = numpy.fabs(fdr_ben-0.05).argmin()

print "Critical s for s_i>s is ", xs[critical_idx]*sigma, 'FDR=%g' % fdr_ben[critical
print "Number of genes is", observed_ben[critical_idx], '(%g)' % (observed_ben[critic
```

```
91 print "Number of remaining genes is", observed_ben[critical_idx]-expected_ben[critica
92
93 pylab.figure(2)
94 pylab.semilogy(xs*sigma,expected_ben,'k-',label='Expected')
95 pylab.semilogy(xs*sigma,observed_ben,'r-',label='Observed')
96 pylab.semilogy(xs*sigma,observed_ben-expected_ben,'b-',label='Observed-Expected')
97 pylab.semilogy([xs[critical_idx]*sigma,xs[critical_idx]*sigma],[1,observed_all[0]],'k
98 pylab.ylim([1,5e03])
99 pylab.xlim([0,0.1])
100 pylab.xlabel('Fitness effect, $s$')
101 pylab.ylabel('Number of genes with $s_i \geq s$')
102 pylab.legend(frameon=False,loc='upper right')
103 pylab.savefig('problem_4_c.pdf')
104
105 pylab.figure(42)
106 pylab.semilogy(xs*sigma,3e-08*(observed_ben-expected_ben),'b-',label='Observed-Expect
107 pylab.semilogy([xs[critical_idx]*sigma,xs[critical_idx]*sigma],[1,observed_all[0]],'k
108 pylab.ylim([1e-08,1e-05])
109 pylab.xlim([0.02,0.1])
110 pylab.xlabel('Fitness effect, $s$')
111 pylab.ylabel('Distribution of fitness effects, $\\int_{s}^{\\infty} U_b \\rho(u) du$'
112 pylab.legend(frameon=False,loc='upper right')
113 pylab.savefig('problem_4_d.pdf')
114
115 #pylab.show()
```