

BSC6883: Evolutionary Game Assay

Jeffrey West¹, 

¹Integrated Mathematical Oncology Department, H. Lee Moffitt Cancer Center & Research Institute

 [@mathoncbro](#)

1 Review: replicator equation

In IMO1, we discussed “evolutionary game theory” wherein individuals (i.e. cancer cells) adhere to a fixed strategy, and replicate in proportion to their strategy’s payoff. We have a population of N cells that can be sensitive (parental cell line), N_S , or resistant N_R , such that $N_S + N_R = N$.

When a subpopulation of S interacts with S, the subpopulation experiences a fitness effect A. When S interacts with R, S experiences fitness effect B and R experiences fitness effect C. Two Rs interact with fitness effect D. These effects are summarized in the game matrix. The row is the strategy of cell type 1, and the column is the strategy of this focal cell’s neighbor, cell type 2.

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (1)$$

This can be represented by an exponential growth model for the number of sensitive (parental) (N_S) and resistant (N_R) cells.

$$\dot{N}_S = N_S \left(a \frac{N_S}{N_S + N_R} + b \frac{N_R}{N_S + N_R} \right) \quad (2)$$

$$\dot{N}_R = N_R \left(c \frac{N_S}{N_S + N_R} + d \frac{N_R}{N_S + N_R} \right) \quad (3)$$

Note: when normalized to track proportion of each cell type over time ($x_1 = \frac{N_S}{N_S + N_R}$ and $x_2 = \frac{N_R}{N_S + N_R}$), this reduces to the replicator equation. Thus, we define “fitness” as the growth rate of each population, given by:

$$f_1 = (Ax)_1 = ax_1 + bx_2 \quad (4)$$

$$f_2 = (Ax)_2 = cx_1 + dx_2 \quad (5)$$

2 Purpose of this lesson

Our goal is to quantify the value of the payoff matrix elements (a, b, c, d) in a biological experimental system of two competing populations: treatment-sensitive ($x_1; f_1$) and treatment-resistant ($x_2; f_2$). To do so, we employ the “evolutionary game assay” which seeds cells at different ratios and quantifies their growth rate.

initial sensitive fraction, $x_1(0)$	initial resistant fraction, $x_2(0)$
1.00	0
0.90	0.10
0.75	0.25
0.50	0.50
0.25	0.75
0.10	0.90
0	1.00

We experimentally measure the growth rate of S and R at each of the initial conditions shown in the table above. The growth rate is interpreted as the cell type's fitness:

$$N_S(t) = N_S(0)e^{f_1 t} \quad (6)$$

$$N_R(t) = N_R(0)e^{f_2 t} \quad (7)$$

In this way, we have an experimentally measured value for fitness of each type as a function of resistant fraction: $f_i(x_2)$. It's convenient to re-write eqns. 4 and 5 as functions of x_2 :

$$f_1(x_2) = (b - a)x_2 + a \quad (8)$$

$$f_2(x_2) = (d - c)x_2 + c \quad (9)$$

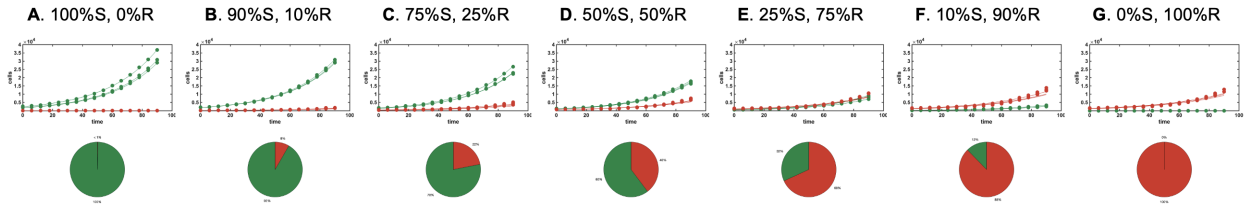


Figure 1

3 Evolutionary Game assay

As seen in figure 1, cells are seeded across a range of initial resistant fraction (see table above), and the growth rate of sensitive and resistant cells is measured for each condition, and repeated for three replicates. This is known as the “evolutionary game assay”¹.

We can use least-squares (Matlab, Python, etc) to estimate the growth rate of sensitive (f_1) and resistant (f_2) cells. With repeated experimental replicates, we have a mean value and standard deviation of growth rates. As seen in figure 2, this gives us the fitness as a function of the number of resistant cells.

3.1 Fitting the fitness functions

Now, we can use least-squares regression to find the best fit for our fitness functions in eqns. 8 and 9. Fitting a line through our data will return estimates for the slope and y-offset of the form:

$$f_1(x_2) = p_1 x_2 + q_1 \quad (10)$$

$$f_2(x_2) = p_2 x_2 + q_2 \quad (11)$$

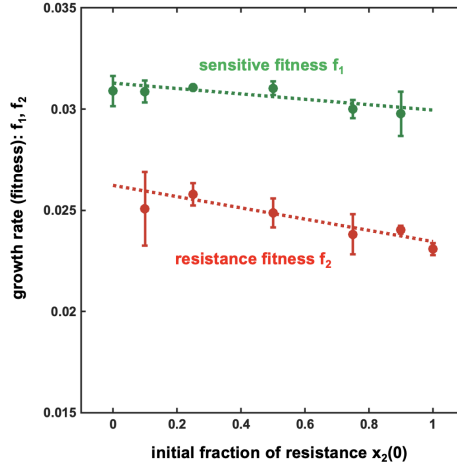


Figure 2

where $p_1 = (b - a)$, $p_2 = (d - c)$, $q_1 = a$, and $q_2 = c$. Often, when fitting a least-squares regression, software packages will return a covariance matrix. Examples of this are `numpy.polyfit` (Python) and `lscov` (Matlab). For two variables, p and q , the covariance matrix is given by:

$$C = \begin{bmatrix} \sigma(p, p) & \sigma(p, q) \\ \sigma(q, p) & \sigma(q, q) \end{bmatrix} \quad (12)$$

The diagonal elements of the matrix represent the variance estimates for p and q , while the off-diagonal elements represent the covariance of q and p (a measure of the joint variability of two random variables). See supplementary section 6.1 for details. Note that $\sigma(p, p) = \sigma_p^2$ is defined as the variance of p , and this matrix is symmetric: $\sigma(p, q) = \sigma(q, p)$.

The fitting process gives us estimates of the mean value of p_i and q_i and the variance of each. **However, we desire estimates of the mean value of variance of the elements of the payoff matrix: a, b, c, d , not estimates of p_i and q_i !** To find these, we must propagate the estimates of error (i.e. variance). This is called error propagation.

4 Error propagation

Propagation of uncertainty (or propagation of error) is the effect of variables' uncertainties (or errors or variance) on the uncertainty of a function based on them. For example, here we have estimates for mean and standard deviation of two random variables: $p_i = \bar{p}_i \pm \sigma_{p_i}$ and $q_i = \bar{q}_i \pm \sigma_{q_i}$. We want to propagate the uncertainty of p_1 and q_1 to derive the uncertainty estimates for payoff elements a, b, c, d . Define these as functions of p_i and q_i :

$$a = q_1 \quad (13)$$

$$b = p_1 + q_1 \quad (14)$$

$$c = q_2 \quad (15)$$

$$d = p_2 + q_2 \quad (16)$$

Error does NOT need to be propagated for a and c , since they are given ($\sigma_{q_1} = \sigma_a$ and $\sigma_{q_2} = \sigma_c$). But b and d are (very simple) function of p_i and q_i . Because the covariance matrix, $C(p, q)$ is known, we can use the following relation. Let $C(X, Y)$ be the covariance of X and Y . We use our error propagation rule:

$$C(\alpha X + \beta Y, \varepsilon W + \gamma V) = \alpha \varepsilon C(X, W) + \alpha \gamma C(X, V) + \beta \varepsilon C(Y, W) + \beta \gamma C(Y, V) \quad (17)$$

where $\alpha, \beta, \varepsilon, \gamma$ are constants.

For example, to find covariance of b , we substitute $\alpha = \beta = \varepsilon = \gamma = 1$ and $X = W = p_1$, $Y = V = q_1$ we arrive at the following equation:

$$C(p_1 + q_1, p_1 + q_1) = C(p_1, p_1) + C(p_1, q_1) + C(q_1, p_1) + C(q_1, q_1) \quad (18)$$

This is equivalent to variance of b such that:

$$\sigma_b^2 = C(b, b) = C(p_1 + q_1, p_1 + q_1) \quad (19)$$

Similarly, to find covariance of d , we substitute $\alpha = \beta = \varepsilon = \gamma = 1$ and $X = W = p_2$, $Y = V = q_2$ we arrive at the following equation:

$$C(p_2 + q_2, p_2 + q_2) = C(p_2, p_2) + C(p_2, q_2) + C(q_2, p_2) + C(q_2, q_2) \quad (20)$$

Similarly, his is equivalent to variance of d such that:

$$\sigma_d^2 = C(d, d) = C(p_2 + q_2, p_2 + q_2) \quad (21)$$

For completeness, we now know the variance of each variable, in terms of our original covariance matrix.

$$Var(a) = C(q_1, q_1) \quad (22)$$

$$Var(b) = C(p_1, p_1) + C(p_1, q_1) + C(q_1, p_1) + C(q_1, q_1) \quad (23)$$

$$Var(c) = C(q_2, q_2) \quad (24)$$

$$Var(d) = C(p_2, p_2) + C(p_2, q_2) + C(q_2, p_2) + C(q_2, q_2) \quad (25)$$

Shown in figure 3 is the mean and standard deviation (square root of variance) for each variable of interest.

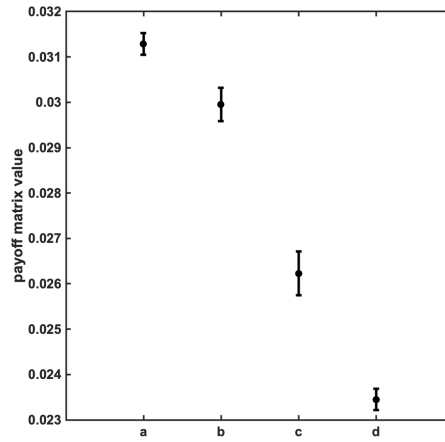


Figure 3

5 Availability of code

All the Matlab code used to generate the plots in these lecture notes can be found online².

References

1. Kaznatcheev, A., Peacock, J., Basanta, D., Marusyk, A. & Scott, J. G. Fibroblasts and alectinib switch the evolutionary games played by non-small cell lung cancer. *Nature Ecology & Evolution* **3**, 450 (2019).
2. West, J. BSC-6883 lecture notes. <https://github.com/JeffreyWest/IMO2> (2023).

6 Supplementary

6.1 Variance and covariance

Let p and q be two random variables with associated mean values, \bar{p} and \bar{q} , and associated variance, σ_p^2 and σ_q^2 . The variance of a random variable is given by:

$$\sigma_p^2 = \frac{1}{n-1} \sum_{i=1}^n (p_i - \bar{p})^2 \quad (26)$$

where n is the number of samples. Similarly, the covariance (a measure of the joint variability of two random variables) can be written: x and y is given by:

$$\sigma(p, q) = \frac{1}{n-1} \sum_{i=1}^n (p_i - \bar{p})(q_i - \bar{q}) \quad (27)$$