**A dynamical systems model for the measurement of cellular senescence**

Daniel Galvis, Darren Walsh, Lorna W. Harries, Eva Latorre, James Rankin

***J. R. Soc. Interface***

**Nomenclature**

Proliferative populations ( for , )

Growth-arrested population ()

Apoptotic population ()

Senescent population ()

Dead (not a population/removed from the system; )

Total population ()

Arbitrary Model Population ()

Total population doublings ()

Rates ()

Markers (, SA--Gal, H2AX, Ki-67, TUNEL. denotes an arbitrary marker)

Number of Passages

**Model equations**

The model is defined by the following system of ordinary differential equations (refer to Fig 1 in the manuscript for a schematic diagram of the possible transitions between populations):

for

*Note:* denotes cells that go through the maximum number of divisions. They are then senescent. denotes cells that jump to senescence when they could still divide one more time.

*Note:* is the rate of transition from population to population .

**Data Fit/Cost Function**

In the optimisation routines the following cost function was used:

,

where is the total number of passages in the experiment and is an index for each passage. is a list of the markers: , SA--Gal, Ki-67, H2AX, and TUNEL.

is the total number of times the population has doubled calculated from cell counts at each passage. All of the other marker values are taken from counting the fraction of cells expressing the marker at each passage.

is derived from the increase in the sum of all populations over time in the model.

SA--Ga is the fraction of cells in the model population over time.

Ki-6 is given by the fraction in the model population over time where is the fraction of the contribution of the growth-arrested population to the marker Ki-67.

H2A is given by the fraction of in the model population over time where is the fraction of the contribution of the growth-arrested population to the marker H2AX and is the fraction of the contribution of the proliferating population to the marker H2AX.

is the fraction of *A* cells in the model population over time.

*Note*: in some cases, we optimize the cost function using a subset of the passages (see Fig 4 in the main manuscript). In this case we define two different error functions, and . is the error from only the included passages; is the error when all passages are included.

**Change in Error**

The change in error for both the re-optimizations and the sensitivity analysis are calculated as follow:

**Parameters/Constraints**

The state number *N =* 50 is the number times a cell in the population can divide (analogous to the Hayflick number). We select 50 rather than optimizing for the value because the change in the error of the cost function changes very little with respect to changes in *N*. We tried *N =* 40, 50, 60, 100, 150. The relative changes in %error are +1.271%, 0.000% (comparing *N =* 50to *N* = 50), -0.377%, -1.101%, -0.071% respectively. The transition rate parameters and are positive numbers determined through optimisation (2 of 14 free parameters). The transition rate parameters ,,, are lists of values with a linear relationship to as set by the positive-valued end points and determined by optimisation (8 of 14 parameters). The fraction of contribution from each population (P or G) to each marker (H2AX or Ki-67) is set by , , , which take values between 0 and 1 are were determined by optimisation (3 of 14 free parameters). These contributions are not known but form part of the optimisation cost function (rather than part of the dynamical system).

**Optimization**

We optimize using the genetic algorithm found in MATLAB2018B. We changed the following options: , and used the default values otherwise. In addition, we also ran the constrained optimization function using the best solution from the genetic algorithm as an initial point. This often would improve the likelihood that the solution would reach a minimum. We increased , and used default settings otherwise.

In an effort to reach the global minimum, we reran the optimization algorithms multiple times for each method. For the initial optimization, where and the full dataset was included, we ran the algorithm 20 times and used the parameter set that minimized the cost function. For every other optimization, such as changing the value of or using a subset of the passages, we ran the algorithm 10 times each with a randomly generated seed population, and used the parameter set that minimized the cost function.

*Note*: In the case where we were optimizing with a subset of passages, we used the parameter set that minimized , not . In some cases, improving the optimization of increased , e.g. when only fitting to a small subset of the data, say the first few points.

**Sensitivity Analysis**

For the sensitivity analysis (Fig 5 in the main manuscript), we changed each parameter while keeping all of the other parameters fixed at the optimized solution. In the case of parameters , we varied both and to create 2-dimensional plots of the change in error.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Parameters |  |  |  |  |  |  |  |  |  |  |
| Rates | 0.0276 | 0.0049 | 0.0035 | 0.0010 | 0.0028 | 0.0053 | 0.0019 | 0.0016 | 0/fixed | 0.0015 |
| Parameters |  |  |  |  |  |  |  |  |  |  |
| Proportions | 0.2060 | 0.9995 | 0.9533 |  |  |  |  |  |  |  |
| Parameters |  |  |  |  |  |  |  |  |  |  |
| Value | 50 |  |  |  |  |  |  |  |  |  |

**Optimization assuming**

We investigated the case where growth-arrested cells cannot transition to senescence, resulting in an significant proportion of growth-arrested cells at the end of the experiment (See Fig S1). Comparing the table of parameters to the original parameter set (Table 1 in the manuscript), it looks like the results are similar, other than that here. Interestingly, both versions of the model suggest that , where as is closer to 1 than in the original model. We cannot draw a strong conclusion on whether the transition from G -> S should be included in the model based on the fit to experimental data. Fig S1, ~20% of cells are growth-arrested at the end and ~80% are senescent, whilst, only increases to 0.8133 from 0.8101 in the original version of the model.

**Parameters affecting time to 85% senescence**

The time it takes to reach 85% senescence was investigated with the model when changing the rate parameters,, , and,  as shown in Fig S2. The figure shows a greyscale map of the change in time to reach 85% senescence depending on the gradient of ,, , and, with respect to increasing . Larger values on the x-axis indicate a larger initial rate and larger values on the y-axis a larger final rate . Black dots indicate the operating values of these parameters for the model. Darker shading indicate that the progression towards senescence is slowed. The transition rates (top left panel) and (bottom right panel) have the greatest effect. As can be expected, decreasing the rate of transition from proliferating to senescent does indeed slow the time to reach 85% senescence. Interestingly, slowing early and increasing later proliferation rates can decrease the rate of transition to senescence.

**Parameter sensitivity with holdout method**

As a further evaluation of the robustness of the model and the relative importance of each parameter, we used the holdout method of cross validation. First, we randomly selected 20 distinct sets of 13 (out of 16) time points of the data (~80% of the data). We reran the optimization on each of these testing sets (repeating each optimization 10 times as described above) to obtain a distribution of errors across testing sets. Note: as with the other subset optimizations, we optimize using . The was then used to calculate a distribution for the percent change in error. Moreover, for this method we also define the error of the points that were omitted from the optimization and the difference in error for the holdout points relative to the baseline error for the same points: . We also found the distribution of percent change in holdout error, which is the defined as:

.

This first results from this analysis is that the meanstd changes in error over the 20 holdout re-optimizations was 1.821.31% across all markers and data points, thus demonstrating that in general the optimization is not sensitive to the arbitrary removal of a small percentage of the data used for fitting. Specifically, for the holdout data points the meanstd error was 15.759.23% with the largest contribution occurring for data points at the beginning or end of the experiments (see also Fig 4 in the main paper).

As another test of parameter sensitivity, we fixed each parameter individually at the best-fit parameter from the full optimization, and reran the optimization on the same 20 sets described above, but with all the other parameters free (table below). This resulted in a % holdout error mean/std for each parameter fixed in isolation. One-sided t-tests were used testing for significant decreases in the holdout error for fixed parameters vs. all parameters free. Decreases in this error imply that fixing a given parameter limits the error introduced when points are held out.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Parameters |  |  |  |  |  |  |  |  |  |  |
| Mean % change holdout error | 16.78 | 16.85 | 16.49 | 15.37 | 15.71 | 14.39 | 15.53 | 14.09 | 16.51 | 13.94 |
| STD % change holdout error | 10.77 | 10.88 | 10.22 | 8.31 | 9.25 | 6.70 | 9.67 | 8.70 | 9.84 | 8.23 |
| P-value for decrease vs. baseline | 0.63 | 0.63 | 0.59 | 0.45 | 0.49 | 0.30 | 0.47 | 0.28 | 0.60 | 0.26 |
| Parameters |  |  |  | baseline |  |  |  |  |  |  |
| Mean % change holdout error | 15.58 | 15.29 | 16.14 | 15.75 |  |  |  |  |  |  |
| STD % change holdout error | 9.69 | 8.91 | 10.44 | 9.23 |  |  |  |  |  |  |
| P-value for decrease vs. baseline | 0.48 | 0.44 | 0.55 | N/A |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

No parameter reached significance for a reduction in error suggesting that the original optimisation approach robustly reached a global minimum and that no parameter was redundant for the optimisation.