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

Code documentation.

./Model\_Final/

This directory contains all the data/code used for *A dynamical systems model for the measurement of cellular senescence*. This directory was created by Daniel Galvis, Darren Walsh, and James Rankin. The data was collected by Eva Latorre and Lorna Harries.

MODEL NOTES: See the manuscript methods section/supplementary materials for more information.

Model Connections:

P0 -> P2 -> ... -> Pn-1 -> Pn/S (cell division, PP)

Pi -> A (apoptosis, PA)

Pi -> G (growth arrest, PG)

Pi -> S (senescence, PS)

A -> D (cell death, AD)

G -> S (GS)

Not Optimized Parameters:

state\_num (n, number of prol) - this is usually set to 50

Optimized Parameters:

LeftPP, RightPP: P0->P1 rate, Pn-1 -> S rate

LeftPA, RightPA: P0->A rate, Pn-1 -> A rate

LeftPG, RightPG: P0->G rate, Pn-1 -> G rate

LeftPS, RightPS: P0->S rate, Pn-1 -> S rate

(note Pi -> X is defined by the line connecting LeftPX and RightPX)

**Pi -> X falls along the line defined by the endpoints P0->X and Pn-1->X**

AD: Transition rate from apoptosis to cell death

GS: Transition rate from growth-arrested to senescent

P\_in\_H2AX (average fraction of proliferative cells expressing H2AX)

G\_in\_Ki67 (average fraction of growth-arrested cells expressed in Ki67)

G\_in\_H2AX (average fraction of growth-arrested cells expressed in H2AX)

Cell Types:

proliferative (prol/P)

growth-arrested (grar/G/GA)

apoptotic (apop/A)

senescent (sene/S)

Note: The population of dead cells (D) is not tracked!

Other Notation:

PD - population doublings

DATA:

**dataset.xlsx** – This contains the original dataset

**create\_data\_file.m** – generates a struct called **data** and outputs it to **comparison\_data.mat**

**data** = *struct with fields:*

**cum\_hours** – array of cumulative hours passed after each passage

**cum\_PD** – array of the cumulative number of population doublings

**pass\_b\_gal** – array of the fraction of cells expressing beta gal at each passage

**pass\_ki\_67** – array of the fraction of cells expressing ki67 at each passage

**pass\_H2AX** – array of the fraction of cells expressing H2AX at each passage

**pass\_tunel** – array of the fraction of cells expressing TUNEL at each passage

(each of these fields has the same number of data points, time is kept by cum\_hours)

COMPUTING:

**matlab\_version\_info.mat** – Contains the versions of all toolboxes that were installed.

RESULT\_FILES:

**\*result\_filename\*.mat** – Contains the result of an optimization as well as the input information for that optimization. The genetic algorithm is run followed by fmincon and the result values that minimize Error\_sub are kept!!

**Error\_sub** – error of the data points that were fit by the optimizer

**Error\_tot** – error of all data points including those not fit by the optimizer

**fit –** a struct similar to data, except that it can only have a subset of the time/data points. These are the points that were fit by the optimization. Same format as data struct.

**options\_fmincon** – fmincon is applied after genetic algorithm. Options here.

**options\_ga** – options for the genetic algorithm

**range\_lb** – lower bounds for the 13 parameters (always 0) algorithm

**range\_ub** – upper bounds for the 13 parameters for genetic algorithm

**range\_lb\_final –** lower bounds for the fmincon optimization (set to 0.5x result of genetic algorithm optimization unless the value doesn’t make sense [e.g. negative])

**range\_ub\_final** – upper bounds for the fmincon optimization (set to 1.5x result of genetic algorithm optimization unless the value doesn’t make sense [e.g. > 100%])

**result** – the best result of the optimization. These are the 13 parameters

**result\_mid** – result of the genetic algorithm only

**seed** – genetic algorithm seed (this might depend on number of cores used unfortunately)

**state\_num** – Number of proliferative states, generally 50 unless otherwise noted.

**parameters\_best\_final.mat – This contains the best optimization of the full system over 21 attempts. Every reoptimization to subsets or change of parameters (sensitivity analysis) is compared to this initial parameter set!!!**

**curves2D.mat** – Trajectory feature information derived from **Run\_curvs2D.m**

**curves\_intercept.mat** –Trajectory feature information derived from **Run\_curv.m**

**curves\_slope.mat** – Trajectory feature information derived from **Run\_curv.m**

**sensitive2D.mat** – Parameter sensitivity information derived from **Run\_sens2D.m**

**sensitive\_intercept.mat** – Parameter sensitivity information derived from **Run\_sens.m**

**sensitive\_slope.mat** – Parameter sensitivity information derived from **Run\_sens.m**

FUNCTIONS (AUXILLARY):

**data\_subset(data, fw\_pts)** – uses **data\_subset\_single(…)** to create the struct called fit, which only has a subset of the points in data. Those are the ones that get fit by the optimizer.

**data\_subset\_single(data.xxx, fw\_pts)** – takes an array of points and uses fw\_pts to determine which points to keep in the array.

**feature\_finder(result, state\_num)** – takes a trajectory and calculates various features. Features are:

* Slope of senescent population at 50% senescent cells (sigmoid approx.)
* Slope of proliferative population at 50% proliferative cells (sigmoid approx.)
* Max value of growth arrested cells
* Time to max value of growth arrested cells
* Max value of apoptotic cells
* Time to max value of apoptotic cells
* Time to 85% senescent

**model\_fun(result, state\_num, end\_time)** – runs the dynamical system with the parameters in result. end\_time is the number of hours to run the simulation.

**ode\_system(t,x,param)** – This is the function that gets used by the ode solver to run the dynamical system.

**opt\_fun(guess, state\_num, fit)** – evaluates the cost function for the points in fit using the parameters in guess.

**run\_optimization(fit, data, range\_lb, range\_ub, state\_num, type, seed\_in, output\_file)** – Runs the genetic algorithm or surrogateopt with fmincon for the given set of inputs and prints **\*result\_filename\*.mat**.

**run\_optimization\_leaveout(leaveout, fit, data, range\_lb, range\_ub, state\_num, type, seed\_in, output\_file)** – same as **run\_optimization(…)** except that it prevents the optimizer from adjusting the parameter with index leaveout during the fmincon optimization. The parameter can still move during ga or surrogateopt unless range\_lb/range\_ub are adjusted so that parameter is fixed.

FUNCTIONS (VISUALIZATION): See Create\_Figures.m for the use of these functions.

Figure 1B (and some extra):

**vis\_param(result, state\_num, data)** – prints Figure 1B and Error\_tot for the given set of result parameters

**vis\_param\_supp(result, state\_num, data, flag)** – prints the output values of the parameters in a slightly different format from vis\_param(…). The flag ‘good’ or ‘bad’ determines the color to plot so that low Error\_tot parameter sets and high Error\_tot parameter sets can be seen simultaneously. (See Extra\_Figures.m)

Figure 3/Supplemental Figure 1 (and some extra):

**vis\_traj(result, state\_num\_data)** – prints Figure 3 and the Error\_tot for the given set of result parameters

**vis\_traj\_supp(result, state\_num, data, flag)** – prints the trajectories over the multiple genetic algorithm optimizations. The flag ‘good’ or ‘bad’ determines the color to plot so that low Error\_tot parameter sets and high Error\_tot parameter sets can be seen simultaneously. (See Extra\_Figures.m)

Figure 4:

**vis\_sene\_traj(result,state\_num,data)** – print only senescent trajectories and data

\*Note: the Figure 4 bar chart code is in Create\_Figures.m

Figure 5:

**vis\_sens2D.m –** This routine visualizes the changes in Error\_tot with respect to changes in each of the parameters. Must run Run\_sens2D.m first. That code does the actual parameter sensitivity analysis. This is called 2D because it shows 2D plots of parameter values for P0 -> X and Pn-1 -> X and allows their values to vary independently.

**vis\_sens(sensitive, labels)** - This routine visualizes the changes in Error\_tot with respect to changes in each of the parameters. Must run Run\_sens.m first. The array **sensitive** is derived from **sensitive\_intercept.mat** or **sensitive\_slope.mat**, which represent different ways in which the parameters P0 -> X and Pn-1 -> X were changed. Since these parameters define a line, either the line was shifted or the slope was changed. See Run\_sens.m for more details.

Supplemental Figure 2 (and some extra):

**vis\_curvs2D(idx) –** This routine visualizes the changes in various features of the trajectories with changes in the values of the parameters. Must run Run\_curv2D.m first. That code does the actual feature analysis analysis. This is called 2D because it shows 2D plots of parameter values for P0 -> X and Pn-1 -> X and allows their values to vary independently. Features are:

* Slope of senescent population at 50% senescent cells (sigmoid approx.) (1)
* Slope of proliferative population at 50% proliferative cells (sigmoid approx.) (2)
* Max value of growth arrested cells (3)
* Time to max value of growth arrested cells (4)
* Max value of apoptotic cells (5)
* Time to max value of apoptotic cells (6)
* Time to 85% senescent (7)

idx is an integer that determines which feature to plot

**vis\_curve\_full(curve,labels)** - This routine visualizes the changes in various features of the trajectories with changes in the values of the parameters. Must run Run\_curv.m first. That code does the actual feature analysis analysis. The array **curve** is derived from **curves\_intercept.mat** or **curves\_slope.mat**, which represent different ways in which the parameters P0 -> X and Pn-1 -> X were changed. Since these parameters define a line, either the line was shifted or the slope was changed. See Run\_curv.m for more details.

**vis\_curve(curve,idx,labels) –** Same as **vis\_curve\_full(…)** except that the idx defines which features is plotted.

Extra Figures:

**vis\_features(result, state\_num) –** This function takes an optimization result and shows how the various features of the trajectories are calculated. Features are:

* Slope of senescent population at 50% senescent cells (sigmoid approx.)
* Slope of proliferative population at 50% proliferative cells (sigmoid approx.)
* Max value of growth arrested cells
* Time to max value of growth arrested cells
* Max value of apoptotic cells
* Time to max value of apoptotic cells
* Time to 85% senescent

**fit\_sigmoid(ins, times, series)** – A cost function for fitting the senescent and proliferative graphs to a sigmoid.

**sigmoid(ins, times)** – A sigmoid function. Ins is the slope and shift.

FUNCTIONS (MASTER):

**Create\_Figures.m** – generates all the figures used in the paper

**Extra\_Figures.m –** generates other figures not used in the paper

**check\_traj.m** – Check the Error\_tot/trajectory figures for changing groups of parameters

**Run\_opts.m** – performs all optimizations used in the paper

1. All point optimizations
2. Remove beginning points
3. Remove end points
4. Remove middle points
5. Skip points
6. GA -> S held at 0
7. Various state\_num values
8. Holdout method (3 points held out randomly)

**Run\_opts\_kfold(input)** – a function for performing optimizations while fixing single parameters at best value.

**Run\_curv.m** – generates **curve\_intercept.mat** and **curve\_slope.mat**. See the routine for more information.

**Run\_curvs2D.m**– Calculates trajectory feature information for changes in all parameters. Each parameter is varied individually, except for P0->X Pn-1->X cases, where a 2D grid of points is determined. Features are:

* Slope of senescent population at 50% senescent cells (sigmoid approx.)
* Slope of proliferative population at 50% proliferative cells (sigmoid approx.)
* Max value of growth arrested cells
* Time to max value of growth arrested cells
* Max value of apoptotic cells
* Time to max value of apoptotic cells
* Time to 85% senescent

**Run\_sens.m** – generates **sensitive\_intercept.mat** and **sensitive\_slope.mat**. See the routine for more information.

**Run\_sens2D.m** –Calculates parameter sensitivity information for changes in all parameters. Each parameter is varied individually, except for P0->X Pn-1->X cases, where a 2D grid of points is determined.

**holdout\_info1.m** – Finds increase in Error\_tot values and statistics for the holdout method. It also checks whether fixing single parameters at best value changes the % increase in Error\_tot for the holdout method.

**holdout\_info2.m** – Finds increase in error of the held out points and statistics for the holdout method. It also checks whether fixing single parameters at best value changes the % increase in held out points for the holdout method.

OTHER NOTES:

result – this array often comes up, and represents a set of 13 parameters for the model. The order of the parameters is always:

['LeftPP', 'RightPP', 'LeftPG', 'RightPG', 'GS', 'LeftPA', 'RightPA', 'LeftPS', 'RightPS', 'AD', 'P\_in\_H2AX', 'G\_in\_Ki67', 'G\_in\_H2AX']

KFold and Holdout are used interchangeably to represent the method where 20% of points are randomly chosen and left out of a model optimization.