Here we explain how we got the related data to plot final figures rather than data itself due to size constraints of git. 1-5 are the data name and we explain how we obtain the data by the following python codes. The explanation of how to execute the code is also attached below.

**1. dgs\_b1\_c1:** generated form the python code main\_b\_and\_c.py.

**2. dgs\_b1\_c1:** generated from the python file main\_dgds.py (Output the distribution of individual survival. In the main code for outputting heatmap data, set the growth rate at the time of individual extinction and death to *-1*. Therefore, the code classifies areas with a growth rate of *-1* and represents them as white regions, while areas with a growth rate other than *-1* are represented as light blue regions).

**3. dgs\_b1\_c1\_small:** same as dgs\_b1\_c1 but with a small scape of dg and ds.

**4. bc\_dg0\_ds0 & bc\_dg0.1\_ds0 & bc\_dg0\_ds0.1 & bc\_dg0.1\_ds0.1 & bc\_dg0.2\_ds0.2 & bc\_dg0.3\_ds0.3:** generated from the python file main\_dgds.py.

**5. Bubble\_A & Bubble\_B & Bubble\_C &. Bubble\_D:** generated from the python code main\_Bubble\_chart.py

**Read me of the file:** main\_b\_and\_c.py

Describe the complete life process of random division in multicellular organisms. We set the mortality rates of germ-like and soma-like cells as constant values ( and ), take the differentiation benefits *b* and differentiation costs *c* as the circulating quantities, repeat the calculation of *k* optimal strategies for each cycle, and finally output *k \* b \* c* (20 \* 30 \* 30) optimal strategies. In the code, *n* represents the number of cell divisions, *portion* represents the proportion of different *dp0*s selected, *M* represents the number of *dp0*s randomly selected for the first time, and *SAM* represents the number of neighbors corresponding to the selected strategy. In each output data, the first *10* rows represent the probability of cell division *10* times (*n=10*), and the probabilities in each column are , , , , , and , respectively. The values in the last row and the first column represent the reproductive rate of the optimal strategy.

**Read me of the file:** main\_dgds.py

We set the cell differentiation benefits b and differentiation costs c as constant values (*b* and *c*), and the mortality rates of g cells and s cells ( and ) as the circulation volume. We repeated the calculation of *k* optimal strategies for each cycle, and finally output *k \* b \* c* (20 \* 30 \* 30) optimal strategies.

**Read me of the file:** main\_Bubble\_chart.py

We set the cell differentiation benefits *b* and differentiation costs *c* as constant values (*b* and *c*), and the mortality rates of germ-like cells and soma-like cells ( and ) as the circulation volume. We repeat the calculation of *k* optimal strategies for each cycle, and finally output *k \* b \* c* (10 \* 13 \* 13) optimal strategies.

**Read me of the file:** data\_Bubble\_chart.py

Filter out all optimal ID strategies through loop statements, then calculate the average total cell count and germ-like cell ratio of all optimal ID strategies under each mortality rate, and take integers.