# **Supporting File 4**. Ontogenic simulation.

We have written a simulation in Matlab which is modeled on cell lineage map by Morrill and Marcus (Morrill and Marcus, 2004). The simulation is an abstract representation of a growing sea urchin embryo. Every node in the map contains a number. This number represents the amount of cells at this point in the lineage. In our simulation each node is represented by an element in a large cell array. The initial embryo in this cell lineage is stored as a zero in a cell array named ‘C’. The first node in Figure (1) is a ‘2’. The simulation represents this ‘2’ as two zeros in a “one by two” matrix stored in the second group of a cell array named ‘C’. C{1} = 0 and C{2} = (0,0). The next node is a ‘4’, represented by a “one by four” matrix of zeros stored in C{3}. The simulation generates these cell arrays using the arrays which correspond to parent cells in the map.

C{3} is represented by the 4 cells after second cleavage. Here the lineage bifurcates and the top half of the bifurcation leads to a 4-this is C{5}-while the bottom half leads to another 4-this is C{6} in the simulation.

The assignment of cell array elements is done like this to keep track of any changes in an array representing the “parent cell”. For example, the parent cell of C{5} and C{6} is C{4}, and we wish to record any changes in the arrays that represent parent cells, while allowing these changes to “propogate” to later generations of cells. The change we speak of will be replacing a single zero in the array with a ‘1’.

We will describe a single iteration of the program below.

At the beginning of the simulation, we generate a random number between 1 and 95,148. This chooses which stage of development will be the setting for an incorporation event (we use empirical data to determine relevant probabilities from **Fig.3B**). The stages are shown in the map as roman numerals at the top of the lineage. If a number is drawn corresponding to stage V, for example, one of the six cell groups in stage V will be selected (uniformly). These six groups are represented in the cell array as C{10} through C{15}. After selecting one of these groups, a single zero in the array will be chosen uniformly. This zero will be changed to a one.

Suppose C{10} was chosen. C{10} corresponds to the cells labeled An11 in the map. C{10} is represented as a “one by four” matrix of zeros i.e. (0,0,0,0). We randomly choose one of these zeros uniformly (each element of C{10} has a probability of 25% to be chosen) and change it to a one, i.e. C{10} now looks like this: (0,1,0,0).

The program generates C{1} then C{2} etc. From our lineage we can see that ‘An11’ is a group of parent cells for An111 and An112. An111 and An112 correspond to C{16} and C{17} respectively. Since C{11} is now (0,1,0,0), C{16} = C{17} = (0,1,0,0). The result is that there are two ‘1’s in the total number of cell arrays generated thus far.

When the cell lineage bifurcates into two groups, like An11 into An111 and An112, we create two new groups with the same array as An11. When there is no bifurcation, and the number of cells doubles from one node to the next, as in 4-An2 to 8-An2, our code concatenates the matrix with itself. For example, if 4-An2 = (1,0,0,0); then 8-An2 will concatenate 4-An2 with 4-An2 resulting in (1,0,0,0,1,0,0,0). The program ends at stage IX colored in blue in figure (1). At this stage, the total number of 1s in all groups of stage IX are summed up. This integer is analogous to the total amount of genomic DNA (gDNA) resulting from this experiment. The initial change of a ‘1’ to a ‘0’ represents the “integration event”. An “integration event” is when foreign DNA (in the form of a putative CRM) is introduced to the embryo (before any cell divisions have occurred). The first element of the cell array to have an element changed to a ‘1’(in this example C{10}) is the “site” of the integration event. The presence of this ‘1’ in the daughter cells of the “site” models the propagation of foreign DNA during Mitosis.

Before we run the simulation, we specify a sub group of cells which will be our “target” cells. These “target” cells represent the cells which will drive expression of the incorporated DNA if it is indeed present. This specification allows us to see the outcome of modelled experiment, given that the “target” cells drive expression. A “target” may be a group of cells such as {An221, An222}. The simulation will sum up the total number of 1s present in the cell array groups which correspond to the “target”. This integer represents cDNA. The goal of our simulation is to mimic the sea urchin experiment, where we will be able to measure cDNA and gDNA. The simulation generates numbers for cDNA and gDNA by randomly choosing a “site” for the integration event, and providing cDNA and gDNA measurements.

Morrill, J. B., Marcus, L., Cell lieage chart through tenth cleavage in the sea urchin Lytechinus variegatus. In: C. A. Ettensohn, G. M. Wessel, G. A. Wray, Eds.), Development of sea urchins, ascidians, and other invertebrate deuterostomes: Experimental approaches, Vol. 74. Elservier, 2004, pp. 840.

**Matlab code:**

%{

BarCode sim for Dr. Jongmin Nam

Sean McQuade and Jongmin Nam, Jan 2015 to Dec 2016

using Sea Urchin Lineage chart from "Development of Sea Urchins, Ascidians, and

other invertebrate Deuterostomes: Experimental Approaches"

Authors: Charles A. Ettensohn, Gary M. Wessel, Gregory A. Wray

Elsevier academic press

% i=cellgroup integrated;

% j=cell in group;

% int = number of integrated cells

% exp = number of expressed cells

%}

%%

%All oblique cell cleavages are coded as horizontal cleavages!

tic

clear %Total\_Targets is a cell array with all targets to be compared

%the simulation loops through this cell array

%figure 1

Total\_Targets{1} = 51;

Total\_Targets{2} = 52;

Total\_Targets{3} = 50;%compare (C) with (A) with (B)

%figure 2

Total\_Targets{4} = 51;%A

Total\_Targets{5} = [51,52];%(AB)

Total\_Targets{6} = 50;%C

Total\_Targets{7} = [51,52];%half A half B

global Number\_of\_Iterations

%%

for a = 1:size(Total\_Targets,2)

%parameters

Number\_of\_Trials = 1;

Number\_of\_Iterations = 10000;

%initialize

C{64}=[]; %preallocate memory for Cell array

Integration\_Stage = zeros(Number\_of\_Iterations,1);

Integrated\_Cells = zeros(Number\_of\_Iterations,1);

Expressed\_Cells = zeros(Number\_of\_Iterations,1);

Ratio\_Exp\_to\_Int = zeros(Number\_of\_Iterations,1);

A\_plot = [0, 0, 0]; %place holder

A\_raw = [0, 0, 0]; %place holder

ROP\_log{Number\_of\_Trials} = 0;

%save file name

path = sprintf('Sim30\_target\_%s.txt',mat2str(Total\_Targets{a}));

if a == 7

path = 'half51, half52, and 50';

end

for b = 1:Number\_of\_Trials

for c = 1:Number\_of\_Iterations

%{

The probability of DNA integration occuring in a stage is modeled on

empiracally obtained profile from Dr. Nam. For a given stage, the probability

of DNA integration into the ith group is proportional to the number of

cells in the ith group.

%}

% create integration profile (this must change for oblique clev

%%

h = randi(95148);

if h == 1 %0th stage, yields ~424 integrated cells

i=1;

m=1;

elseif h >= 2 && h<=31 %Stage I, yields ~212 integrated cells

i=2;

m=2;

elseif h>=32 && h<=958 %Stage II, yields ~106 integrated cells

i=3;

m=3;

elseif h>=959 && h<=9589 %Stage III, yields ~64 integrated cells

i=4;

m=4;

elseif h>=9590 && h<=28880 %Stage IV, yields ~32-42 integrated cells

i=randi([5,8]);

m=5;

elseif h>=28881 && h<=46066 %Stage V, yields ~16 integrated cells

p=randi(6);

if p==1

i=10;

elseif p==2

i=11;

elseif p==3 || p==4

i = 12;

elseif p==5 || p==6

i=13;

end

m=6;

elseif h>=46067 && h<=58496 %Stage VI, yields ~8-10 integrated cells

p=randi(14);

if p==1

i=9;

elseif p==2

i=14;

elseif p==3

i = 16;

elseif p==4

i=17;

elseif p==5 || p==6

i=18;

elseif p>=7 && p<=10

i=19;

elseif p==11 || p==12

i=20;

elseif p==13 || p==14

i=21;

end

m=7;

elseif h>=58497 && h<=68321 %Stage VII, yields ~4 integrated cells

p=randi(26);

if p==1

i=22;

elseif p==2

i=23;

elseif p==3 || p==4

i=24;

elseif p==5 || p==6

i=25;

elseif p>=7 && p<=10

i=26;

elseif p>=11 && p<=14

i=27;

elseif p>=15 && p<=18

i=28;

elseif p>=19 && p<=22

i=29;

elseif p>=23 && p<=26

i=30;

end

m=8;

elseif h>=68322 && h<75147 %Stage VIII, yields ~2 integrated cells

p=randi(53);

if p == 1

i = 15;

elseif p>=2 && p<=5

i=randi([31,34]);

elseif p==6 || p==7

i = 35;

elseif p==8 || p==9

i = 36;

elseif p>=10 && p<=13

i = 37;

elseif p>=14 && p<=37

i=randi([38,40]);

elseif p>=38 && p<=53

i=randi([41,44]);

%{

if p==1

i=31;

elseif p==2

i=32;

elseif p==3

i=33;

elseif p==4

i=34;

elseif p==5 || p==6

i=35;

elseif p==7 || p==8

i=36;

elseif p>=9 && p<=16

i=37;

elseif p>=17 && p<=24

i=38;

elseif p>=25 && p<=32

i=39;

elseif p>=33 && p<=40

i=40;

elseif p>=40 && p<=43

i=41;

elseif p>=44 && p<=47

i=42;

elseif p>=48 && p<=51

i=43;

elseif p>=52 && p<=55

i=44;

%}

end

m=9;

elseif h>=75148 && h<=95148 %Stage IX, yields 1 integrated cell

i=randi([45,64]);

m=10;

end

%%

%make vector for hist: Integration\_Stage

Integration\_Stage(c)=m;

% Cell in group C{i} chosen uniformly.

if i == 1

DNA\_integrated\_cell = 1;

elseif i == 2

DNA\_integrated\_cell = randi(2);

elseif i>=3 && i<=11 || i>=14 && i<=17 || i>=31 && i<=34 || i>=45 && i<=46 || ...

i>=22 && i<=23

DNA\_integrated\_cell = randi(4);

elseif i>=12 && i<=13 || i==18 || i>=20 && i<=21 || i>=24 && i<=25 || ...

i>=35 && i<=36 || i>=47 && i<=48 || i>=61

DNA\_integrated\_cell = randi(8);

elseif i==19 || i>=26 && i<=30 || i>=41 && i<=44 || i==49

DNA\_integrated\_cell = randi(16);

elseif i>=37 && i<=40 || i>=50 && i<=60

DNA\_integrated\_cell = randi(32);

end

%the DNA is integrated in the jth cell in the ith group.

j = DNA\_integrated\_cell;

%% The cells multiply with incorporated dna expressed as 1, no incorporation is 0.

%, they multiply either by concatenating a group with itself

%(no bifurcation on the cell lineage) , or by setting two new groups equal

%to the parent group (bifurcation in the cell lineage)

Cells0 = zeros(1,1);

C{1}= Cells0;

C{i}(1)=1;

C{2}= cat(2,C{1},C{1});

C{i}(j)=1;

C{3}= cat(2,C{2},C{2});

C{i}(j)=1;

C{4}= C{3};

C{5}= C{3};

C{i}(j)=1;

C{6}= C{4};

C{7}= C{4};

C{8}= C{5};

C{9}= C{5};

C{i}(j)=1;

C{10} = C{6};

C{11} = C{6};

C{12} = cat(2,C{7},C{7});

C{13} = cat(2,C{8},C{8});

C{14} = C{9};

C{15} = C{9};

C{i}(j)=1;

C{16} = C{10};

C{17} = C{10};

C{18} = cat(2,C{11},C{11});

C{19} = cat(2,C{12},C{12});

C{20} = C{13};

C{21} = C{13};

C{22} = C{14};

C{23} = C{14};

%116 Total Cells

C{i}(j)=1;

C{24}= cat(2,C{16},C{16});

C{25}= cat(2,C{17},C{17});

C{26}= cat(2,C{18},C{18});

C{27}= C{19};

C{28}= C{19};

C{29}= cat(2,C{20},C{20});

C{30}= cat(2,C{21},C{21});

C{31}= C{22};

C{32}= C{22};

C{33}= C{23};

C{34}= C{23};

%232 total cells

C{i}(j)=1;

C{35}= C{24};

C{36}= C{24};

C{37}= cat(2,C{25},C{25});

C{38}= cat(2,C{26},C{26});

C{39}= cat(2,C{27},C{27});

C{40}= cat(2,C{28},C{28});

C{41}= C{29};

C{42}= C{29};

C{43}= C{30};

C{44}= C{30};

C{45}= C{15};

C{46}= C{15};

%424 total cells

C{i}(j)=1;

C{47}= C{35};

C{48}= C{35};

C{49}= cat(2,C{36},C{36});

C{50}= cat(2,C{37},C{37});

C{51}= C{38};

C{52}= C{38};

C{53}= C{39};

C{54}= C{39};

C{55}= C{40};

C{56}= C{40};

C{57}= cat(2,C{41},C{41});

C{58}= cat(2,C{42},C{42});

C{59}= cat(2,C{43},C{43});

C{60}= cat(2,C{44},C{44});

C{61}= cat(2,C{31},C{31});

C{62}= cat(2,C{32},C{32});

C{63}= cat(2,C{33},C{33});

C{64}= cat(2,C{34},C{34});

C{i}(j)=1;

%%

%Count integrated cells within specified "target" groups

%for target = [C,hA, hB], we sum the groups

if a == 7

halfTargetgroup1 = C{Total\_Targets{a}(1)}(1:16);

halfTargetgroup2 = C{Total\_Targets{a}(2)}(1:16);

Expressed\_Cells(c) = nnz(halfTargetgroup1)+ nnz(halfTargetgroup2);

else

for d=1:size(Total\_Targets{a},2)

Expressed\_Cells(c) = Expressed\_Cells(c) + nnz(C{Total\_Targets{a}(d)});

end

end

%Counts integrated cells in all stage 9 groups(also C{45} and C{46} (SM))

Integrated\_Cells(c) = nnz(C{47}) + nnz(C{48}) + nnz(C{49}) + ...

nnz(C{50}) + nnz(C{51}) + nnz(C{52}) + nnz(C{53}) + nnz(C{54}) +...

nnz(C{55}) + nnz(C{56}) + nnz(C{57}) + nnz(C{58}) + nnz(C{59}) +...

nnz(C{60}) + nnz(C{61}) + nnz(C{62}) + nnz(C{63}) + nnz(C{64}) +...

nnz(C{45}) + nnz(C{46}); %these last two terms add SM1 and SM2

%log the ratio of this particular iteration in the experiment

Ratio\_Exp\_to\_Int(c) = Expressed\_Cells(c)/Integrated\_Cells(c);

A\_raw = [A\_raw; Expressed\_Cells(c), Integrated\_Cells(c), Ratio\_Exp\_to\_Int(c)];

%%%%%%%%%%%% if Integrated\_Cells(c) >= 4%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%% A\_plot = [A\_plot; Expressed\_Cells(c), Integrated\_Cells(c), Ratio\_Exp\_to\_Int(c)];

%%%%%%%%%%%% end

A\_plot = [A\_plot; Expressed\_Cells(c), Integrated\_Cells(c), Ratio\_Exp\_to\_Int(c)];

end

%%

%Data is ordered by column 3 (expression ratio), then put into descending

B\_raw = sortrows(A\_raw,3); %orders array with respect to(WRT) col 3

D\_raw = flip(B\_raw,1); %puts into descending order WRT col 3

D\_raw(end,:) = []; %removes the last row from array, initial zeros for A.

CRM\_raw = D\_raw;

%from (eq2)

avg\_exp = sum(CRM\_raw(:,3))/Number\_of\_Iterations; %instinctual calculation

%avg\_exp = sum(CRM\_raw(:,1))/sum(CRM\_raw(:,2)); %paper calculation eq. (2)

B\_plot = sortrows(A\_plot,3); %orders array with respect to(WRT) col 3

D\_plot = flip(B\_plot,1) ; %puts into descending order WRT col 3

D\_plot(end,:) = []; %removes the last row from array, initial zeros for A.

CRM\_plot = D\_plot;

simple\_vector = (1:Number\_of\_Iterations);

% title1 = sprintf('expression profile, Target=%s',mat2str(Total\_Targets{a}));

% figure;

% plot(simple\_vector, Trimmed\_info(:,3))

% xlabel('raw number of trials');

% ylabel({'Rank-Ordered, raw expression','(cDNA/gDNA)'});

% title(title1)

% axis([0, size(Trimmed\_info, 1), 0, 1])

% figure;

% plot(simple\_vector, CRM\_plot(:,3)/avg\_exp,'LineWidth',2)

% xlabel('Rank','fontsize',18);

% ylabel('Expression Level / Mean','fontsize',18);

% figure\_title = sprintf('%s',mat2str(Total\_Targets{a})');

% title(figure\_title)

% axis auto

%%

avg\_expLOG(a)=avg\_exp;

CRMlog{a}=CRM\_plot;

%%

%write information to file

fileID = fopen(path,'a');

fprintf(fileID,'%s%s\t%s\t%i\t%s\t%i\t%s\t%f\n','#Target\_',mat2str(Total\_Targets{a}),'cDNAall'...

,sum(CRM\_raw(:,1)),'gDNAall',sum(CRM\_raw(:,2)),'c/g\_all', avg\_exp);

fclose(fileID);

%write raw data

for h = 1:size(CRM\_raw, 1)

fileID = fopen(path,'a');

fprintf(fileID,'\t%s\t%i\t%s\t%i\t%s\t%f\n','cDNA'...

,CRM\_raw(h,1),'gDNA',CRM\_raw(h,2),'c/g',CRM\_raw(h,3));

fclose(fileID);

end

end

end

figure(1);

plot(simple\_vector, CRMlog{1}(:,3)/avg\_expLOG(1),simple\_vector, CRMlog{2}(:,3)/avg\_expLOG(2),'--',simple\_vector, CRMlog{3}(:,3)/avg\_expLOG(3),':','LineWidth',2)

xlabel('Rank','fontsize',18);

ylabel('Expression Level / Mean','fontsize',18);

title('Comparison of Rank Order Profiles','fontsize',18)

legend('A', 'B', 'C')

DistanceC\_A=rop\_metric(CRMlog{1},CRMlog{2});

DistanceA\_B=rop\_metric(CRMlog{2},CRMlog{3});

DistanceC\_B=rop\_metric(CRMlog{1},CRMlog{3});

inset1a = sprintf('D(A,C) = %3.3f',DistanceC\_A);

inset1b = sprintf('D(A,B) = %3.3f',DistanceA\_B);

inset1c = sprintf('D(B,C) = %3.3f',DistanceC\_B);

text(7500,10,inset1a)

text(7500,9,inset1b)

text(7500,8,inset1c)

%legend('C', 'A', 'B', 'A and B', 'hA and hB')

figure(2);

plot(simple\_vector, CRMlog{4}(:,3)/avg\_expLOG(4),simple\_vector, CRMlog{5}(:,3)/avg\_expLOG(5),'--',simple\_vector, CRMlog{6}(:,3)/avg\_expLOG(6),'-.',simple\_vector, CRMlog{7}(:,3)/avg\_expLOG(7),':','LineWidth',2)

xlabel('Rank','fontsize',18);

ylabel('Expression Level / Mean','fontsize',18);

title('Comparison of Rank Order Profiles','fontsize',18)

legend('A', 'AB', 'C','hAhB') %(CRMlog{4},CRMlog{5},CRMlog{6},CRMlog{7})

DistanceA\_AB=rop\_metric(CRMlog{4},CRMlog{5});

DistanceA\_C=rop\_metric(CRMlog{4},CRMlog{6});

DistanceA\_hAhB=rop\_metric(CRMlog{4},CRMlog{7});

DistanceAB\_C = rop\_metric(CRMlog{5},CRMlog{6});

DistanceAB\_hAhB = rop\_metric(CRMlog{5},CRMlog{7});

DistanceC\_hAhB = rop\_metric(CRMlog{5},CRMlog{6});

inset2a = sprintf('D(A,AB) = %3.3f',DistanceA\_AB);

inset2b = sprintf('D(A,C) = %3.3f',DistanceA\_C);

inset2c = sprintf('D(A,hAhB) = %3.3f',DistanceA\_hAhB);

inset2d = sprintf('D(AB,C) = %3.3f',DistanceAB\_C);

inset2e = sprintf('D(AB,hAhB) = %3.3f',DistanceAB\_hAhB);

inset2f = sprintf('D(C,hAhB) = %3.3f',DistanceC\_hAhB);

text(7000,10,inset2a)

text(7000,9,inset2b)

text(7000,8,inset2c)

text(7000,7,inset2d)

text(7000,6,inset2e)

text(7000,5,inset2f)

toc