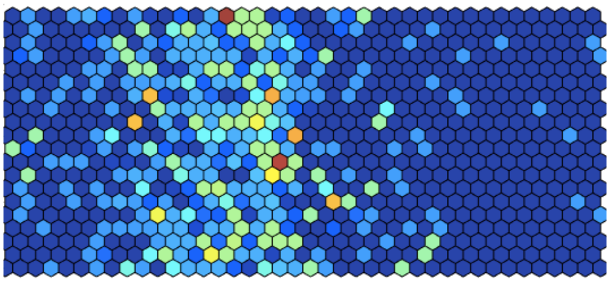
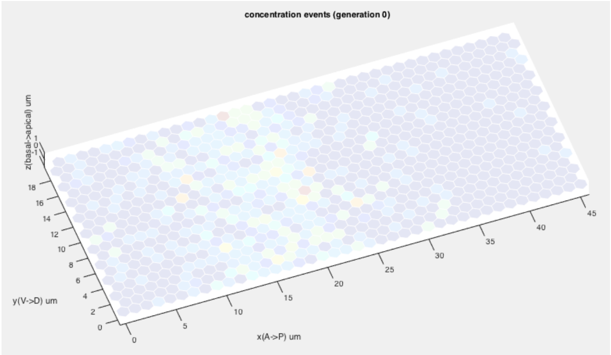
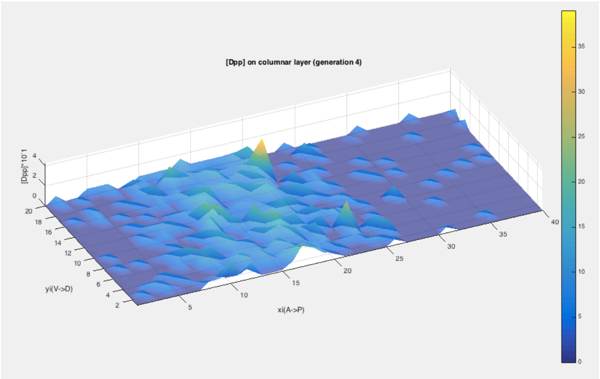
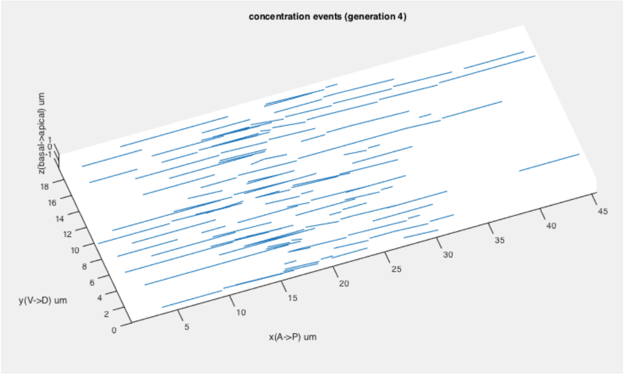
Guild to to run create3Dhex_dir to build cytoneme simulation model









Outline of this document

This document introduces the code structure of create3Dhex_dir.m matlab file which contains everything you will need to build cytoneme simulation model and follow-up analysis. Here is the outline of this guild including:

1. How to simulate the first cytoneme model
2. Code structure
3. Setup parameters
4. Main functions
5. simulation results

1. How to simulate the first cytoneme model

To run the first cytoneme model:

step 1: change the parameter values in function p=parameter_set() (line 50)

For example 1:

To test different hypothesis cytoneme model, you can change:
p.AN, p.LE, p.PF, p.MA

For example 2:

To change wing disc physical model, you can change:
p.nx, p.ny, p.w_dppcent

For example 3:

To choice morphogen, you can change:
p.morphogen_init (Dpp =1, Hh=2), p.amount

For example 4:

To display or save figures, you can change:
p.img, p.show_every_img, p.save_every_img

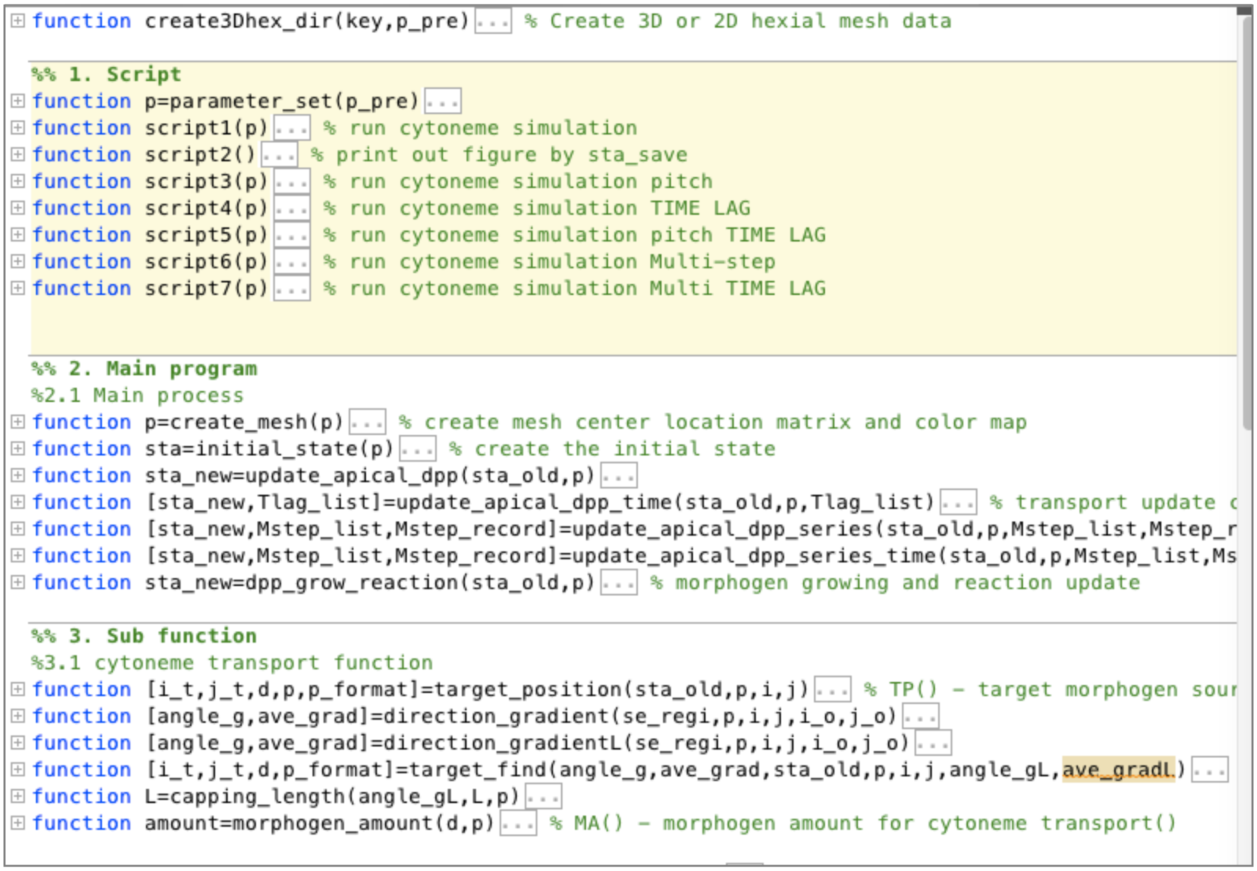
step 2: >> create3Dhex_dir(1)   (type in matlab command line and ENTER)

step 3: finished!

All results will be stored in cytoneme.mat including p(parameter), re_txt(explanations of re), sta_all(all generation results), sta_save(generation results you displayed), re('number of cytonemes in distance range groups';'#cytoneme events';'# cytonemes';'mean cytoneme distances’).

2. Code structure

This is what you will see when you collapse the create3Dhex_dir.m file. There are totally 51 functions in this single matlab file.



```
function create3Dhex_dir(key,p_pre)... % Create 3D or 2D hexial mesh data

%% 1. Script
function p=parameter_set(p_pre)...
function script1(p)... % run cytoneme simulation
function script2()... % print out figure by sta_save
function script3(p)... % run cytoneme simulation pitch
function script4(p)... % run cytoneme simulation TIME LAG
function script5(p)... % run cytoneme simulation pitch TIME LAG
function script6(p)... % run cytoneme simulation Multi-step
function script7(p)... % run cytoneme simulation Multi TIME LAG


%% 2. Main program
%2.1 Main process
function p=create_mesh(p)... % create mesh center location matrix and color map
function sta=initial_state(p)... % create the initial state
function sta_new=update_apical_dpp(sta_old,p)...
function [sta_new,Tlag_list]=update_apical_dpp_time(sta_old,p,Tlag_list)... % transport update c
function [sta_new,Mstep_list,Mstep_record]=update_apical_dpp_series(sta_old,p,Mstep_list,Mstep_r
function [sta_new,Mstep_list,Mstep_record]=update_apical_dpp_series_time(sta_old,p,Mstep_list,Ms
function sta_new=dpp_grow_reaction(sta_old,p)... % morphogen growing and reaction update

%% 3. Sub function
%3.1 cytoneme transport function
function [i_t,j_t,d,p,p_format]=target_position(sta_old,p,i,j)... % TP() - target morphogen sour
function [angle_g,ave_grad]=direction_gradient(se_regi,p,i,j,i_o,j_o)...
function [angle_g,ave_grad]=direction_gradientL(se_regi,p,i,j,i_o,j_o)...
function [i_t,j_t,d,p_format]=target_find(angle_g,ave_grad,sta_old,p,i,j,angle_gL,ave_gradL)...
function L=capping_length(angle_gL,L,p)...
function amount=morphogen_amount(d,p)... % MA() - morphogen amount for cytoneme transport()
```

The first function create3Dhex_dir(key, p_pre) is the main function of this cytoneme simulation model. You can choice which script you want to run using the first input: key. p_pre is the per-parameter variable. If you want to run the model using default parameters setup, you don't need to input p_pre. You can change the default model parameters in the second function parameter_set() before running the model. Script1~7 are script functions.

Functions in ''2. main program'' are important core functions. create_mesh() will create wing disc hexagonally epithelial cell physical model, initial_state() will initiate the morphogen concentration distribution and the model. Update_apical_dpp() will calculate the morphogen transport status.

Example:
To build one cytoneme simulation model and run it, you can type this in matlab command line and run:

```
> create3Dhex_dir(1);
```

To batch run several cytoneme simulation models:

```
> create3Dhex_dir(3)
```

3. Setup parameters

You can change the default parameter values in p=parameter_set() function. p is a matlab structure variable which contains all model parameters. Basically this is the only place you will need to change in this code. Here are some important parameters to run cytoneme simulation listed below:

Cytoneme transport hypothesis model

parameter name	default value	parameter definition	description of parameter content
p.AN	1	pathfinding angle function	PA = 1. const pathfinding angle 2. random pathfinding angle 3. max averaged gradient
p.LE	2	pathfinding length function	PL= 1. const pathfinding length 2. random pathfinding length 3. averaged gradient function 4. highest concentration in thita 5. function of inverse gradient
p.PF	1	probability of formation function	TH= 1. const probability 2. PF const/length, 3. function of averaged gradient, 4. d-effect 5. inverse gradient 6. it jt concentration
p.MA	1	morphogen transport amount function	1. one cytoneme 2. N cytoneme
p.CA	1	cytoneme capping criteria	

Cytoneme transport parameters

parameter name	default value	parameter definition	description of parameter content
p.amount	3	morphogen transport amount	transport morphogen through cytoneme [C/sec] (## no data!!)
p.theta	1	cytoneme grow angle	p.theta=1~12(degree: 0~330) if angle fun=1
p.length	10	cytoneme length function	length of const cytoneme value if length fun=1 (## Dpp_avg:20um [242], Hh_avg: 27um in P, 13um in A [26])
p.prob_format	0.2	cytoneme formation probability	probability of cytoneme formation. if prob fun =1 (## estimated!)
p.stop_format	0.2	cytoneme stop probability	probability of cytoneme stop. if prob fun =1 (## estimated!)
p.gradient_L	10	cytoneme length gradient coefficient	if length fun=3, length=p.gradient_L* averaged gradient(## estimated!)
p.gradient_P2	20	cytoneme probability gradient coefficient	if probability fun=2, p_format=p.gradient_P*agrad; (## estimated!)
p.gradient_P3	10	cytoneme probability gradient coefficient	if probability fun=3, p_format=p.gradient_P*agrad; (## estimated!)
p.gradient_P4	50	cytoneme probability gradient coefficient.	if probability fun=3, p_format=p.gradient_P*agrad; (## estimated!)
p.gradient_P5	1	cytoneme probability gradient coefficient.	if probability fun=3, p_format=p.gradient_P*agrad; (## estimated!)
p.dfun_option	2	d-effect options	if probability fun=4 (## estimated)
p.amount_dist	[0,5,10,20;5,3,2,1]	cytoneme distribution- transport distribution	cutoneme number depend on distance: [4 3 2 1] (## estimated from [26])
p.capping	1	filipodia capping check	
p.max_gener	10	max generation	
p.morphogen_init	1	Morphogen type	morphogen: 1=dpp, 2=Hh
p.pitch	1	number of batch run	script 3 repeat times

Dpp cytoneme parameters

parameter name	default value	parameter definition	description of parameter content
p.prod_init	0	max initial morphogen concentration [C]	(## Dpp=0, Hh=0)
p.prod_time	3.98	morphogen production m/t [C/sec]	(## 3.98 mol/um/s)
p.deg_time	2.52*10^(-4)	morphogen degradationn m/t [C/sec]	(## Dpp 2.52*10-4 sec-1 [20])
p.cyto_rate	6	cytoneme grow rate or transport morphogen rate [L/sec]	(## 5-7 um/sec for Dpp [190])

Hh cytoneme parameters

parameter name	default value	parameter definition	description of parameter content
p.prod_init	0	max initial morphogen concentration [C]	(## Dpp=0, Hh=0)
p.prod_time	0.5	morphogen production m/t [C/sec]	
p.deg_time	3.3*10^(-3)	morphogen degradationn m/t [C/sec]	(## Hh 3.3*10-3 s-1 [201])
p.cyto_rate	6	cytoneme grow rate or transport morphogen rate [L/sec]	(## 5-7 um/sec for Dpp [190])

Mesh parameters

parameter name	default value	parameter definition	description of parameter content
p.nx	40	Drosophila wing disc x size (unit: cell number)	real wing disc nx=120(167um) (ref: actual width: 223(300um)[242])
p.ny	20	Drosophila wing disc y size (unit: cell number)	real wing disc ny=50 (83um) (ref: actual width: 111(150um)[242])
p.r	0.67	hexagonally epithelial cell radius (unit: um)	real radius 0.67um (ref: actual diameter: 0.5-2.2um[242])

p.w_dppcent	8	Dpp producing center width (unit: cell number)	Dpp producing center width=9cells (ref: Dpp signaling center wide: 8~10 cells width)
p.height	0	mesh height	if h=-1 means 2D
p.sta_save_period	10	figure save options	figures and record every generations
p.area_rx	2	cytoneme search area	search space x radius (ref: Dpp: 40um =2*Dpp_avg [242],Hh:54um in P,28um in A [26])
p.area_ry	0	cytoneme search area	search space y radius (ref: Dpp: 40um =2*Dpp_avg [242],Hh:54um in P,28um in A [26])

Figure parameters

parameter name	default value	parameter definition	description of parameter content
p.img	1	figures show and save option	0 means no save and no display result figures
p.save_every_img	0	save result figures	output figures for each display step. 1: output figures, 0: no output figures
p.show_every_img	1	show result figures	show figures for each display step. 1: output figures, 0: no output figures
p.col_range		colormap	Matlab jet colormap range
p.alpha1	0.8	figure1 transparency	alpha value for figure1
p.alpha2	0.7	figure2 transparency	alpha value for figure2
p.range_n	5	cytoneme display option	cytoneme distance table value number
p.range_d	5	cytoneme display option	cytoneme distance table diff distance
p.fig_bound	20	cytoneme display option	cytoneme show boundary

4. Main functions

p=create_mesh(p)

This function creates the wing disc hexagonally epithelial cell 2D and 3D physical model with user-defined sizes and properties which you can define in mesh parameters. For example: wing disc size = p.nx(=40) * p.ny(=20) means this simulated wing disc contains 800 hexagonally epithelial cells.
Related parameters: p.nx, p.ny, p.r, p.w_dppcent, p.height

sta=initial_state(p)

initial_state function output the initial morphogen concentration on Drosophila wing disc at time = 0. In Dpp morphogen case, Dpp is produced from morphogen-producing cells in Dpp producing center region which controlled by parameter p.w_dppcent. This function will output matlab struct variable sta. sta.dpp is the morphogen concentration distribution on wing disc, and sta.cyto is cytoneme on wing disc.
Related parameters: p.nx, p.ny, p.morphogen_init, p.prod_init, p.w_dppcent

sta=Update_apical_dpp(sta,p)

This function calculate and update the cytoneme growing and morphogen transport in every generation.
Step 1: sub function dpp_grow_reaction(sta_old,p) will calculate the morphogen producing and the morphogen reaction with receptor for every cells and update the variable sta.

Step 2: In every cell in every time generation, function target_position will calculate the probability to grow cytoneme, and select the potential target cell which cytoneme will grow and extend from this cell to that target cell.

Step 3: We assume cytoneme-mediated transport is viewed as a stochastic process. If a random number between 0 to 1 is larger than cytoneme formation probability we get from the previous step, cytoneme will extend from this cell to target cell. If not, there is no cytoneme growth.

step 4: If cytoneme growth is happened, amount=morphogen_amount(d,p) will decide how much morphogen will transport from morphogen-producing cell to morphogen-receiving cell.

step 5: calculation process will be applied to every cells on wind disc, and then update and output new dpp concentration distribution and cytoneme status.

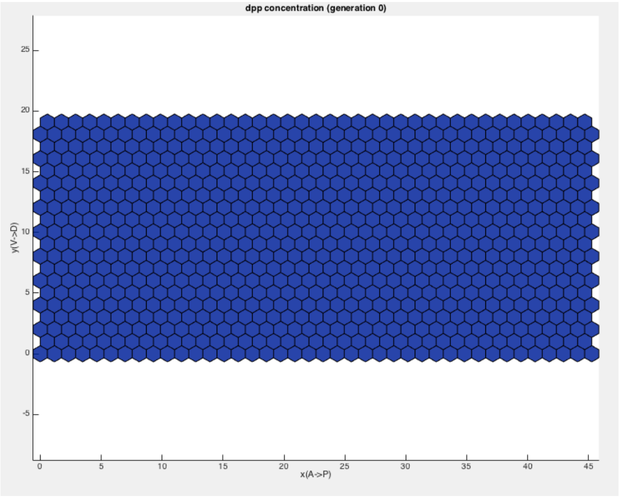
[sta,p]=show_figure(sta,p,ge,sta_save)

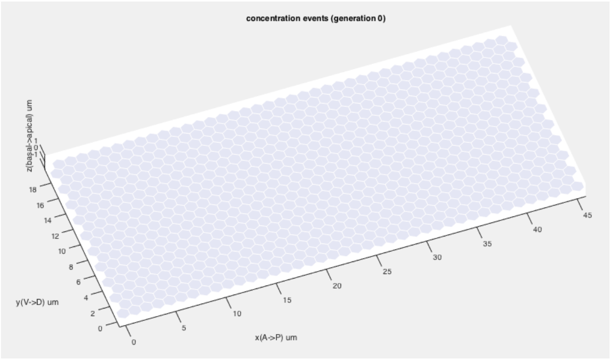
This function will display and save result figures. If p.img =0, then the code will not run this function. If p.show_every_img = 1, it will display every figures in the running process from t=0 to t=max generation (p.max_gener). If p.save_every_img = 1, it will save every figures in the running process from t=0 to t=max generation

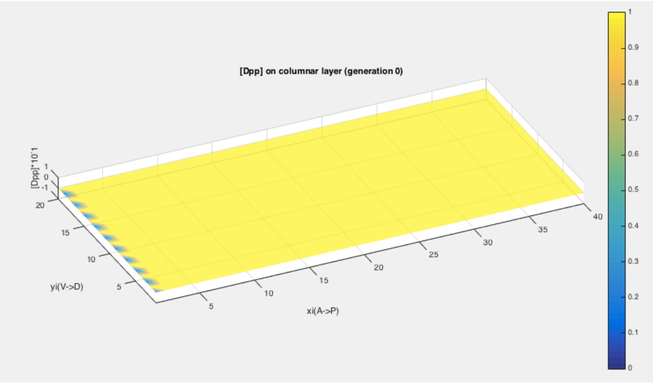
5. Simulation results

Here are result figures you will get once you run the first example model:

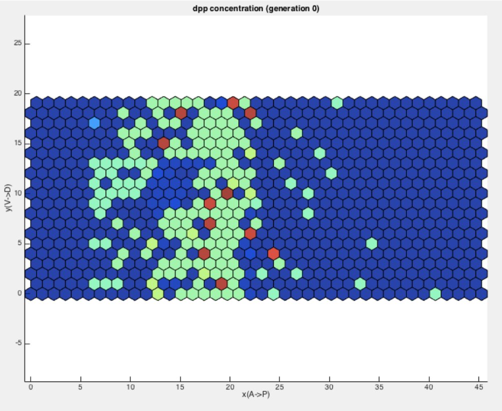
t = 0

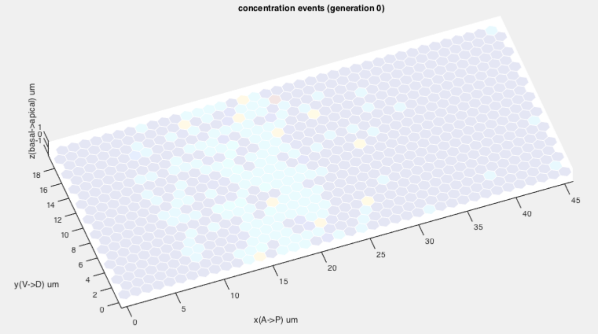


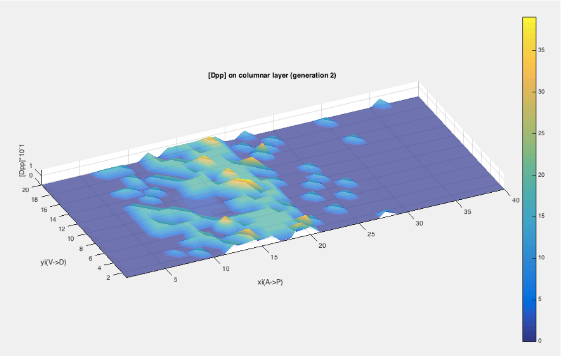


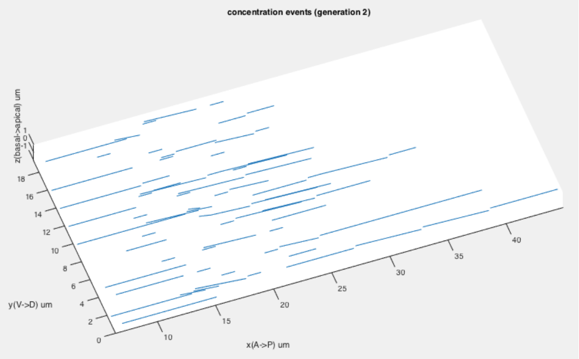


t = 2

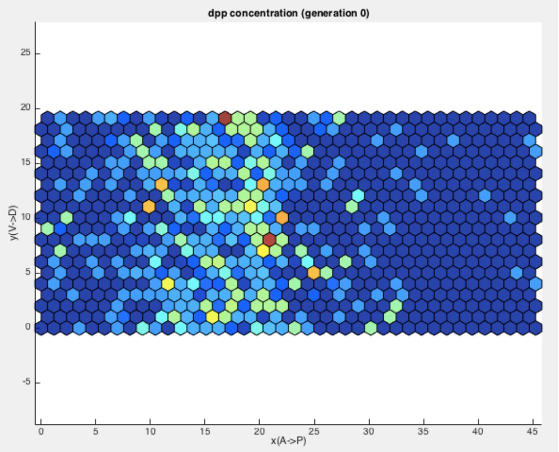

dpp concentration (generation 0)
y(V->D)
x(A->P)
25
20
15
10
5
0
-5
0
5
10
15
20
25
30
35
40
45

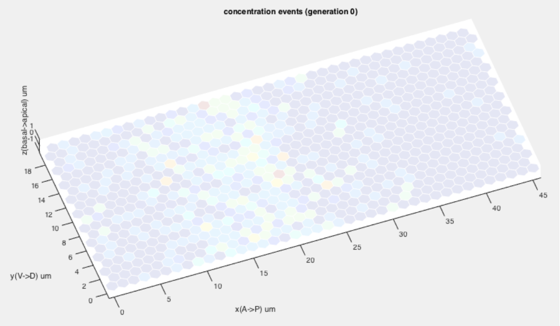

concentration events (generation 0)
z(basal->apical) um
y(V->D) um
x(A->P) um
0
5
10
15
20
25
30
35
40
45

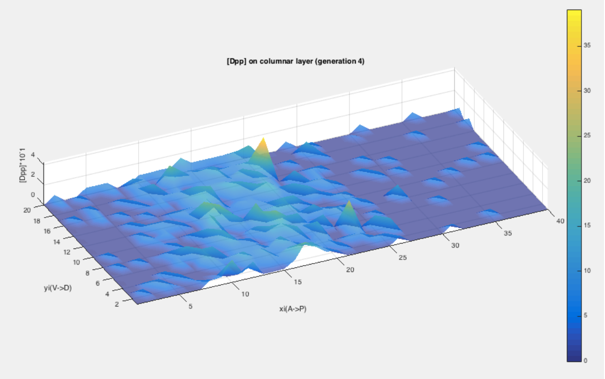

[Dpp] on columnar layer (generation 2)
[Dpp]*10^1
yi(V->D)
xi(A->P)
5
10
15
20
25
30
35
40
0
5
10
15
20
25
30
35

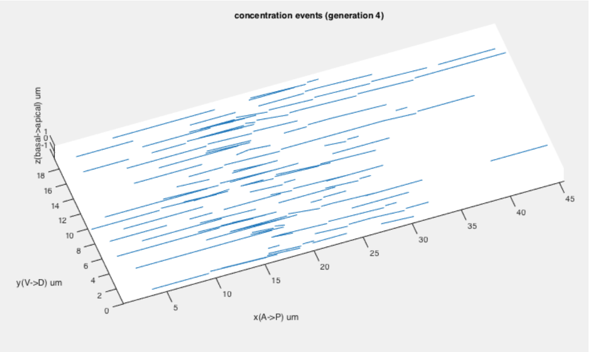

concentration events (generation 2)
z(basal->apical) um
y(V->D) um
x(A->P) um
10
15
20
25
30
35
40

t = 4

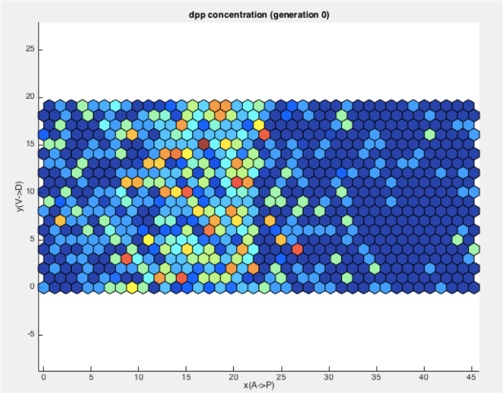

dpp concentration (generation 0)
y(V->D)
x(A->P)
25
20
15
10
5
0
-5
0
5
10
15
20
25
30
35
40
45

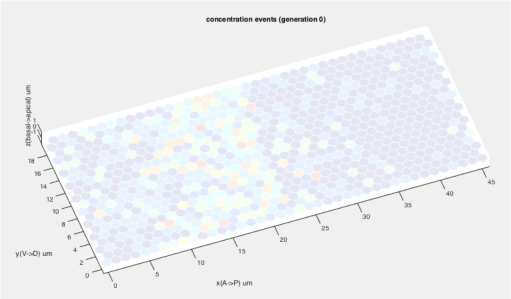

concentration events (generation 0)
z(basal->apical) um
y(V->D) um
x(A->P) um
0
5
10
15
20
25
30
35
40
45

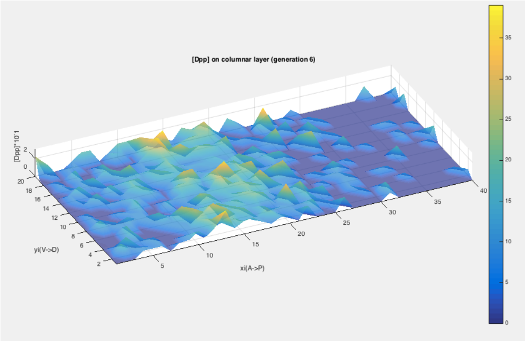

[Dpp] on columnar layer (generation 4)
[Dpp]*10^1
yi(V->D)
xi(A->P)
5
10
15
20
25
30
35
40
0
5
10
15
20
25
30
35

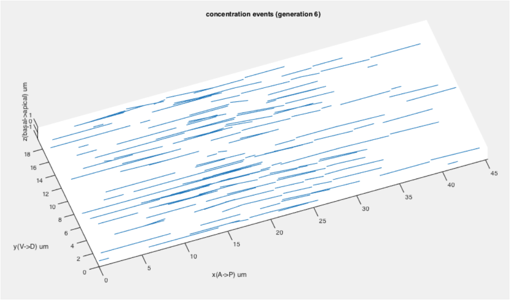

concentration events (generation 4)
z(basal->apical) um
y(V->D) um
x(A->P) um
5
10
15
20
25
30
35
40
45

t = 6

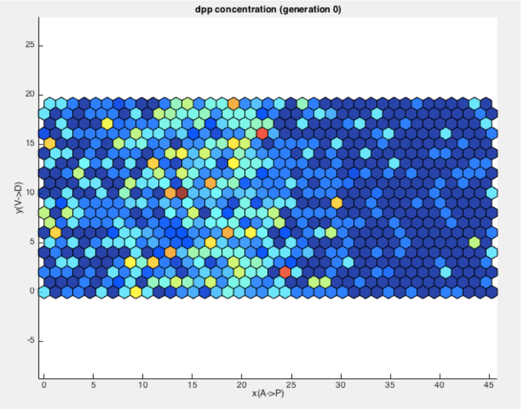

dpp concentration (generation 0)
y(V->D)
x(A->P)
25
20
15
10
5
0
-5
0
5
10
15
20
25
30
35
40
45

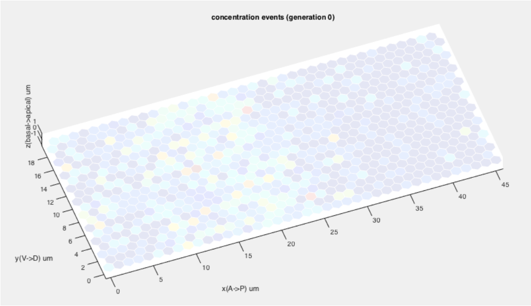

concentration events (generation 0)
z(basal->apical) um
y(V->D) um
x(A->P) um

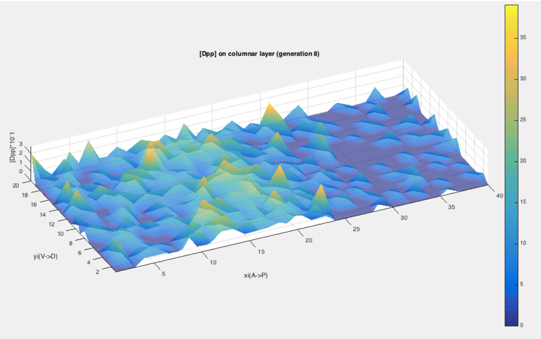

[Dpp] on columnar layer (generation 6)
[Dpp]*10^1
yi(V->D)
xi(A->P)
35
30
25
20
15
10
5
0

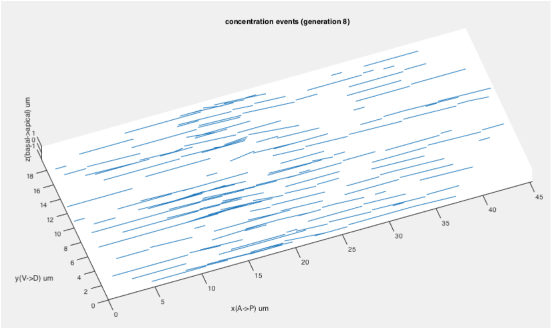

concentration events (generation 6)
z(basal->apical) um
y(V->D) um
x(A->P) um

t = 8

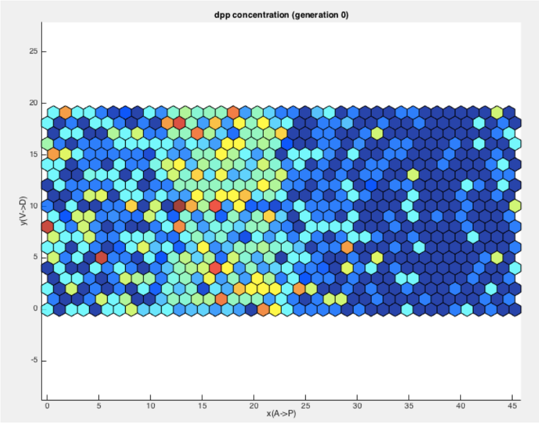

dpp concentration (generation 0)
y(V->D)
x(A->P)
25
20
15
10
5
0
-5
0
5
10
15
20
25
30
35
40
45


concentration events (generation 0)
z(basal->apical) um
y(V->D) um
x(A->P) um

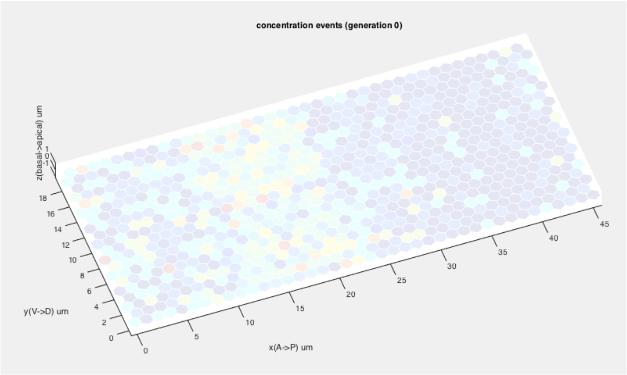

[Dpp] on columnar layer (generation 8)
[Dpp]*10^1
yi(V->D)
xi(A->P)
35
30
25
20
15
10
5
0


concentration events (generation 8)
z(basal->apical) um
y(V->D) um
x(A->P) um

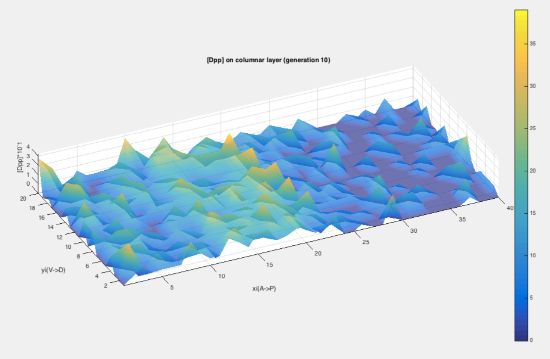
t = 10



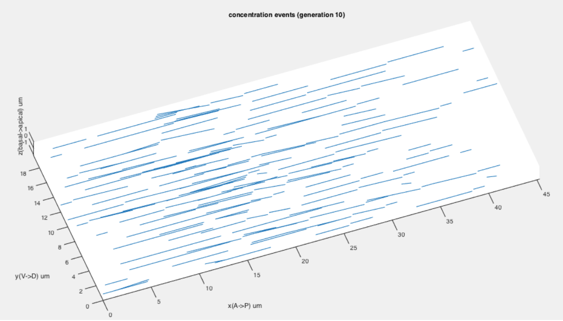
dpp concentration (generation 0)
y(V->D)
25
20
15
10
5
0
-5
0
5
10
15
20
25
30
35
40
45
x(A->P)



concentration events (generation 0)
z(basal->apical) um
1
0
-1
18
16
14
12
10
8
6
4
2
0
y(V->D) um
0
5
10
15
20
25
30
35
40
45
x(A->P) um



[Dpp] on columnar layer (generation 10)
[Dpp]*10^1
4
3
2
1
0
20
18
16
14
12
10
8
6
4
2
yi(V->D)
5
10
15
20
25
30
35
40
xi(A->P)
35
30
25
20
15
10
5
0



concentration events (generation 10)
z(basal->apical) um
1
0
-1
18
16
14
12
10
8
6
4
2
0
y(V->D) um
0
5
10
15
20
25
30
35
40
45
x(A->P) um