Slides, videos, links and more:
https://github.com/physicell-training/ws2022

Session 7: Cell Interactions

Paul Macklin, Ph.D.
@MathCancer

PhysiCell Project

July 26, 2022

Goals

- Further introduce **cell interactions**
  - Advanced chemotaxis, phagocytosis, effector attack, fusion, transformation

- Extend the **tumor-immune** example
  - Macrophages with M1-M2 axis:
    - release pro-inflammatory in high O2, anti-inflammatory in low O2
  - CD8+ T cells:
    - Follow pro-inflammatory signals, attack tumor cells, inhibited by anti-inflammatory
  - Damaged tumor cells die

- Begin **farmer-prey-zombie** example
  - Farmers
    - create food for prey
  - Zombies:
    - Seek and attack prey, eat the dead, merge into super-zombies
  - Prey:
    - aggregate, seek food, reproduce, avoid and counter-attack zombies
    - damage causes transformation into zombies

Advanced interactions

Biased random migration & "standard" chemotaxis

Biased random migration

- **Key parameters:**
  - s (instantaneous) speed
  - $\boldsymbol{d}_{\text{bias}}$ preferred migration direction
  - b preference for choosing to migrate along $\boldsymbol{d}_{\text{bias}}$
  - $\boldsymbol{d}_{\text{motility}}$ direction of motility
  - $T_{\text{persistence}}$ mean time between choosing new migration directions
- **Choosing a migration direction:**
  - Let $\boldsymbol{\theta}$ be a random unit vector. Then

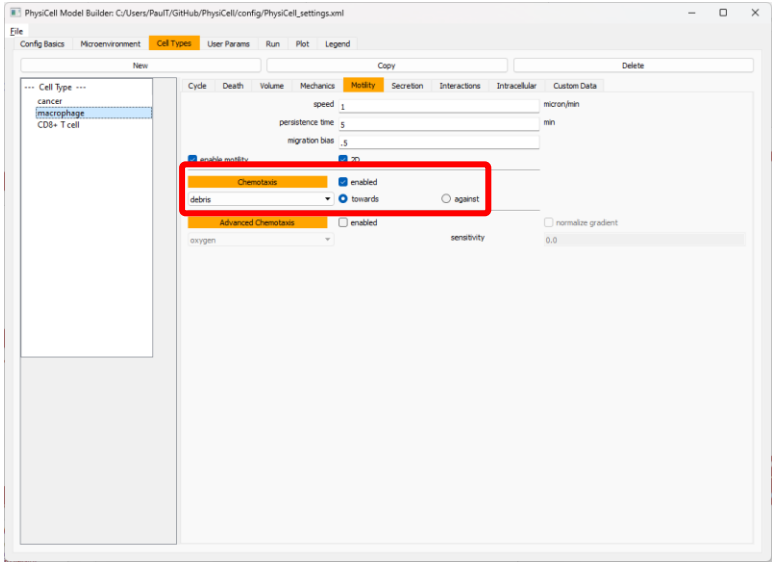
$$\boldsymbol{d}_{\text{motility}} = b \cdot \boldsymbol{d}_{\text{bias}} + (1 - b) \cdot \boldsymbol{\theta}$$

- **Contribution to cell velocity**

$$\boldsymbol{v} = s \frac{\boldsymbol{d}_{\text{motility}}}{|\boldsymbol{d}_{\text{motility}}|} + \text{other terms } \ldots$$

"Standard" chemotaxis

- Set $\boldsymbol{d}_{\text{bias}} = \frac{\nabla \rho}{|\nabla \rho|}$ for some substrate ρ



Set motility parameters and **enable motility**

Under **chemotaxis**

- choose which substrate in the drop-down
- choose towards / away the gradient
- check **enabled**

Advanced chemotaxis

"Advanced" chemotaxis

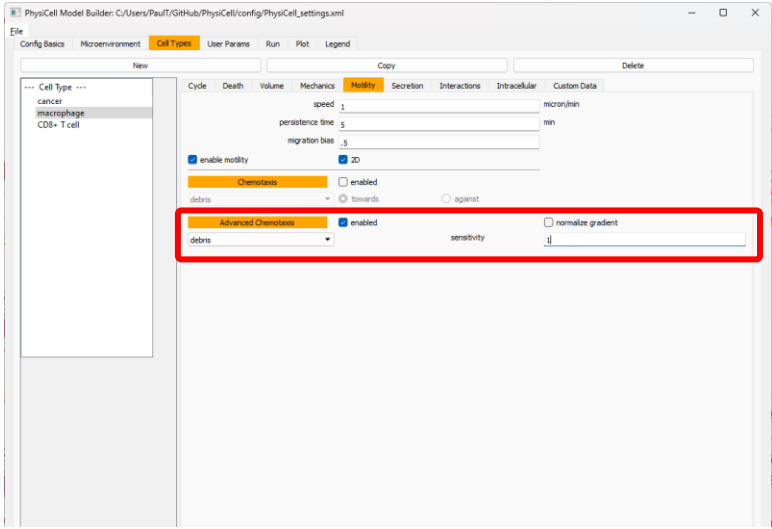
- Linear combination of gradients for more nuanced movement

$$\boldsymbol{d}_{\text{bias}} = \sum_j w_j \nabla \rho_j$$

  - if $w_j > 0$, then move along the gradient
  - if $w_j < 0$, then move against the gradient

- normalized option:

$$\boldsymbol{d}_{\text{bias}} = \sum_j w_j \frac{\nabla \rho_j}{\left|\nabla \rho_j\right|}$$



Set motility parameters and **enable motility**
Under **advanced chemotaxis**

- Set the weight for each substrate
- positive weights move along gradient
- negative weights move against gradient
- check **enabled**

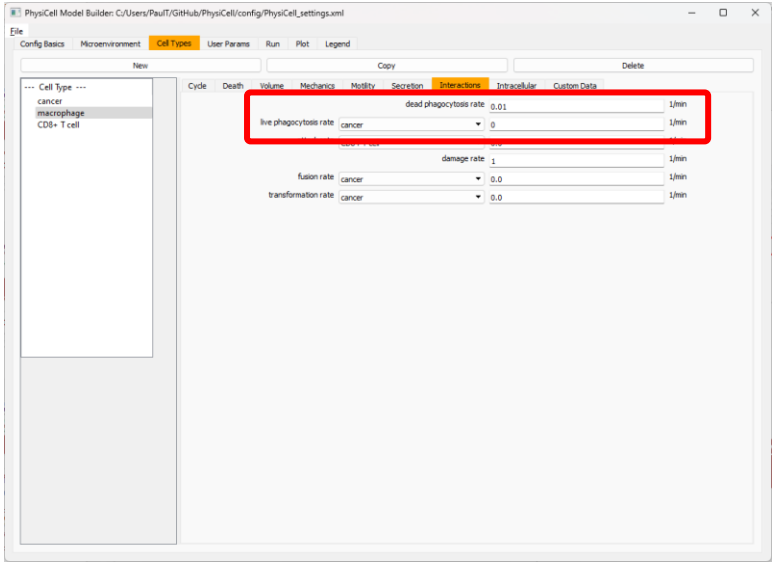
Neighbor lists and interactions (1)

Neighbors list

- Every cell has a list `pCell->state.neighbors`
  - pointers to all cells within interaction distance
  - automatically updated by mechanics
  - We can use it to automate more common interactions with nearby cells

Phagocytosis

- **Key parameters:**
  - $r_{\text{phago,dead}}$ rate of phagocytosing dead cells
  - $r_{\text{phago},i}$ rate of phagocytosing live cells of type *i*
- **Implementation:**
  - If a neighbor is dead, phagocytose with probability $r_{\text{phago,dead}}\ \Delta t$
  - If a neighbor is live with type *i*, phagocytose with probability $r_{\text{phago},i}\ \Delta t$
  - Absorb fluid contents into fluid
  - Absorb solid contents into cytoplasmic solids
  - Internalized substrates are added to the phagocytosing cell
  - Shrink towards target volume
  - **Note:** Processed every $\Delta t_{\text{mechanics}}$



Under **interactions**

- set the dead cell phagocytosis rate
- choose a cell type from the drop-down and set its live phagocytosis rate

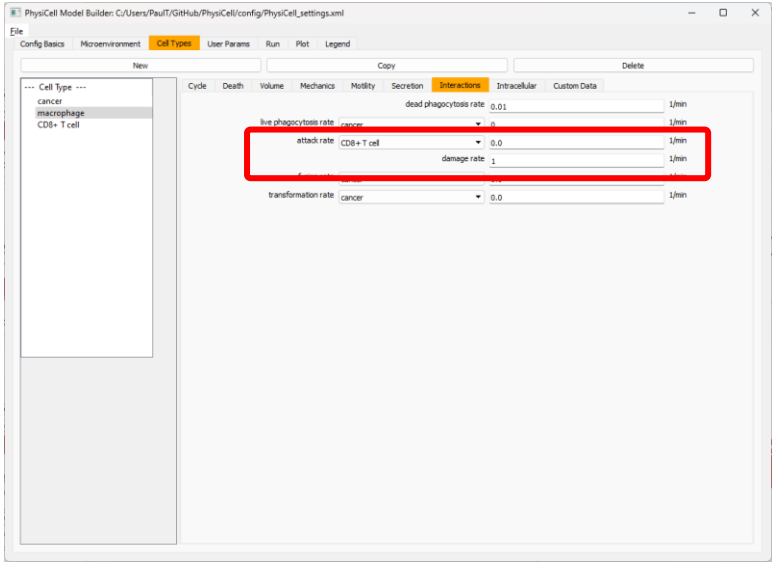
Neighbor lists and interactions (2)

(Effector) attack

- **Key parameters:**
  - $r_{\text{attack},i}$ rate of attacking live cells of type *i*
  - r_{damage} rate of increasing damage during an attack.

- **Implementation:**
  - If a neighbor is live with type *i*, attack with probability $r_{\text{attack},i}\ \Delta t$
  - If attacking, increase cell's damage by $r_{\text{damage}}\Delta t$

  - **Note:** You need a C++ rule (and additional hypothesis) for the attack to cause death.
  - **Note:** Processed every $\Delta t_{\text{mechanics}}$
  - **Note:** Currently we only attack one cell per time step
  - **Note:** Multiple cells can attack a cell concurrently
  - **Note:** If left unchanged, then a cell's **damage** is the total cumulative attack time on a cell. (AUC)



Under **interactions**

- choose a cell type from the drop-down and set its live phagocytosis rate
- set the attack rate

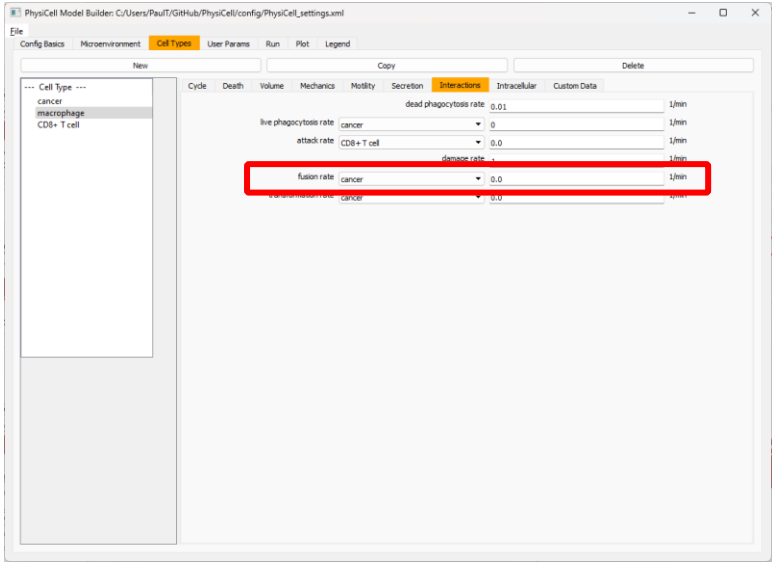
Neighbor lists and interactions (3)

Fusion

- **Key parameters:**
  - $r_{\text{fusion},i}$ rate of fusing with live cells of type *i*

- **Implementation:**
  - If a neighbor is live with type *i*, fuse with probability $r_{\text{fuse},i}\ \Delta t$
  - Combine volumes (and sub-volumes)
  - Combine internalized substrates
  - New position is at center of volume
  - Combine nuclei (tracked in `state.number_of_nuclei`)

  - **Note:** Processed every $\Delta t_{\text{mechanics}}$
  - **Note:** Currently we only fuse one cell per time step



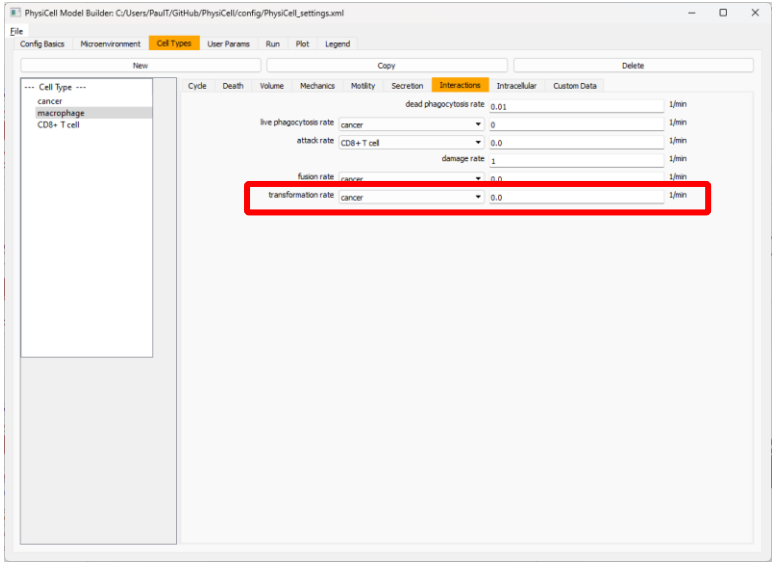
Under **interactions**

- choose a cell type from the drop-down and set its fusion rate

Neighbor lists and interactions (4)

Transformation

- **Examples:**
  - differentiation, mutation, trans-differentiation
- **Key parameters:**
  - $r_{\text{transform},i}$ rate of transforming into cell type *i*
- **Implementation:**
  - Probability of changing to cell type *i* is $r_{\text{transform},i}\,\Delta t$
  - Use the cell definition of type *i* to reinitialize the cell
    - ♦ Phenotype, custom data, and functions overwritten
    - ♦ Could use some future refinement?
      - » preserve internalized substrates?
      - » preserve **conservd** custom variables?
      - » preserve volumes?
  - **Note:** Processed every Δt_{cell}
  - **Note:** Only one transformation per time step
  - **Note:** Could probably be more delicate



Under **interactions**

- choose a cell type from the drop-down and set its transformation rate

Let's extend our previous model!

- Macrophages with M1-M2 axis:
  - ♦ secrete pro-inflammatory factor in except in low O2
  - ♦ secrete anti-inflammatory in low O2

- CD8+ T cells:
  - ♦ Chemotaxis towards pro-inflammatory signals
  - ♦ Attack cancer cells
  - ♦ Anti-inflammatory signals reduce chemotaxis and attack

- Cancer cells:
  - ♦ Damage increases (apoptotic) death

Checklist

- Plan ☐
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☐
  - Add CD8+ T cells ☐
  - Update macrophages ☐
  - Update cancer cells ☐
- Refinement (in C++):
  - Update cancer cell phenotype ☐
  - Create macrophage phenotype ☐
  - Create CD8+ Tcell phenotype ☐
  - Assign functions ☐
  - Compile and test ☐

Planning (1)

- Tumor cell apoptosis varies with damage

$$d = d_0 + (d_{\max} - d_0) H(D)$$

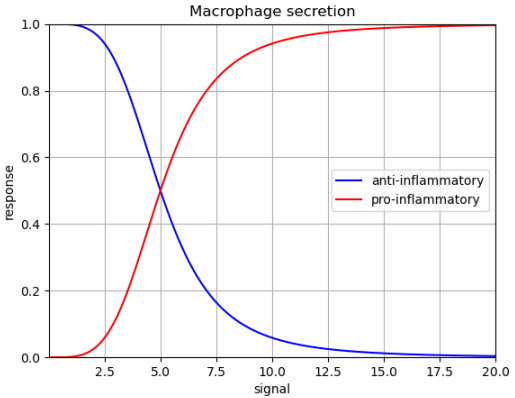
  - $d_{\max} = 100\, d_0$
  - half-max $= 180$ min
  - Hill power $= 2$;

- Macrophage secretion varies with O2

$$S_{\text{pro}} = S_{\text{pro},0}\, H(\sigma)$$

$$S_{\text{anti}} = S_{\text{anti},0} \left(1 - H(\sigma)\right)$$

  - half-max: 5 mmHg
  - Hill power: 4 (for a sharp transition)



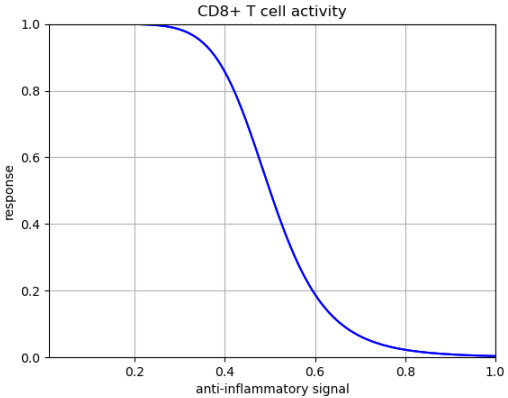
Planning (2)

- CD8+ T cell behavior varies with anti-inflammatory signal

$$s = s_0\left(1 - H(\text{anti})\right)$$

$$r_{\text{attack}} = r_{\text{attack},0}\left(1 - H(\text{anti})\right)$$

  - half-max $= 0.5$
  - Hill-power $= 8$ (VERY steep response)



- **Note:** These parameters are *not calibrated*. Chosen for convenience!

Planning (3)

- Microenvironment
  - Pro-inflammatory marker with diffusion constant 1000, decay 1 (length scale: 100)
    - no-flux boundary conditions
  - Anti-inflammatory marker with diffusion constant 1000, decay 1 (length scale: 100)
    - no-flux boundary conditions

- Additional custom cell data (known once you have planned your cell functions)
  - damage_halfmax (for cancer cell death response to damage)
  - damage_hillpower (for cancer cell death response to damage)
  - damage_relative_max_death (for cancer cell death response to damage)
  - M1M2_halfmax (for macrophage secretion response)
  - M1M2_hillpower (for macrophage secretion response)
  - CD8_halfmax (for CD8+ T cell migration and attack response)
  - CD8_hillpower (for CD8+ T cell migration and attack response)

- Cell definitions
  - CD8+ T cell

Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☐
  - Add CD8+ T cells ☐
  - Update macrophages ☐
  - Update cancer cells ☐
- Refinement (in C++):
  - Update cancer cell phenotype ☐
  - Create macrophage phenotype ☐
  - Create CD8+ Tcell phenotype ☐
  - Assign functions ☐
  - Compile and test ☐

Start modeling!

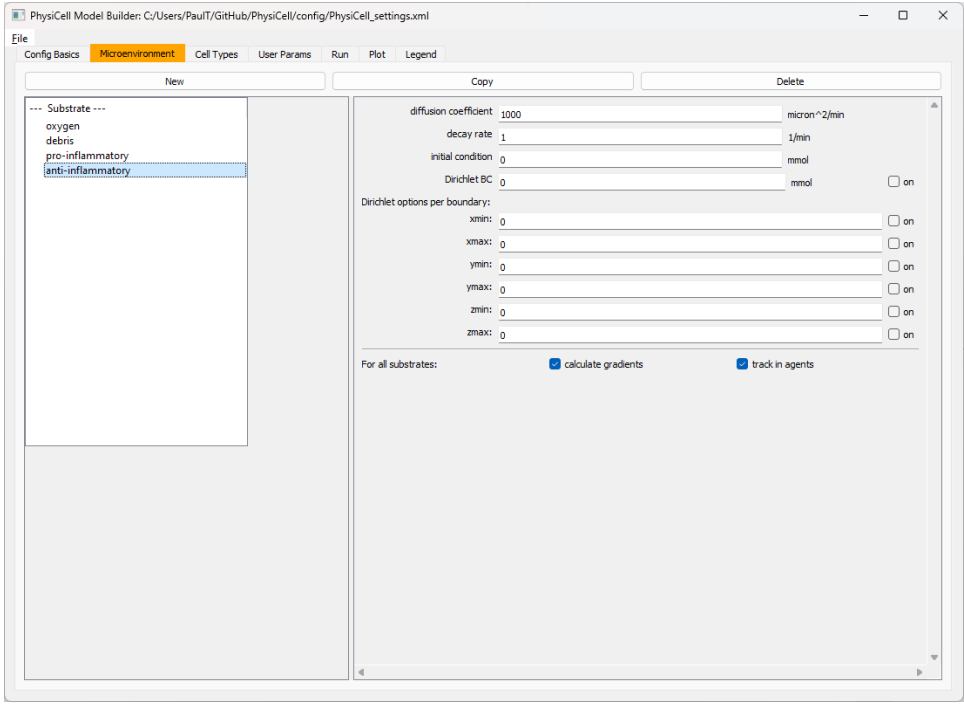
- Continue editing the code from Session 6

- (re)Open Model Builder GUI
  - `python ../PhysiCell-model-builder/bin/pmb.py --studio`

- Open config/PhysiCell_settings.xml, and save.

Edit the model: microenvironment

- Go to **microenvironment** tab
  - double-click **debris** and **copy**
    - rename it **pro-inflammatory**
    - set **diffusion** to **1000**
    - set **decay** to **1**
    - set **initial condition** to 0
    - disenable the Dirichlet BC
  - select **pro-inflammatory** and **copy**
    - rename it to **anti-inflammatory**

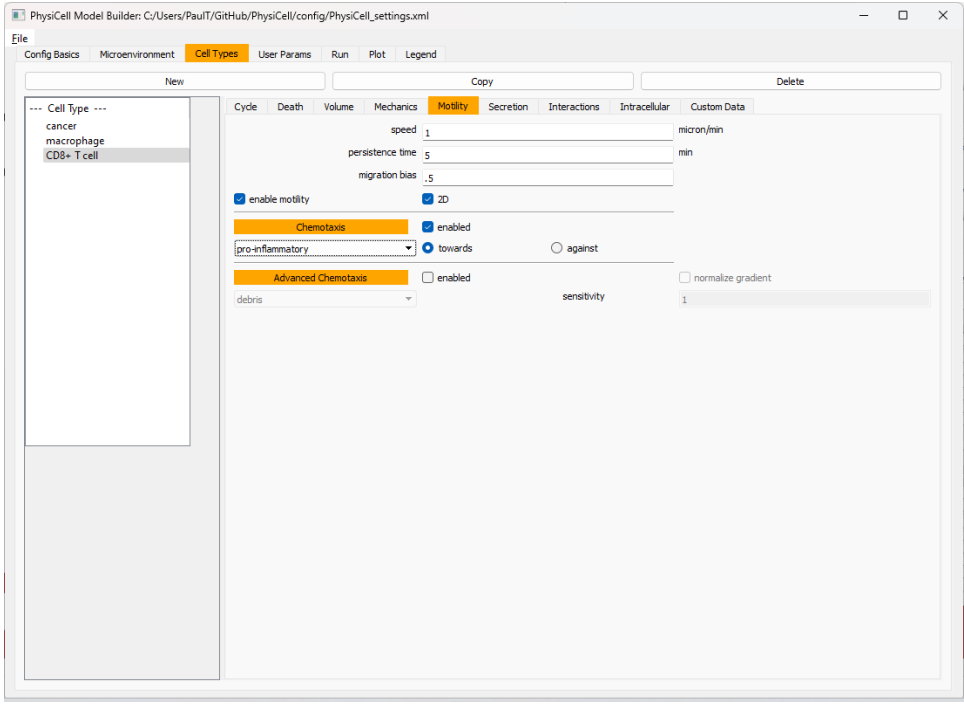


Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☐
  - Update macrophages ☐
  - Update cancer cells ☐
- Refinement (in C++):
  - Update cancer cell phenotype ☐
  - Create macrophage phenotype ☐
  - Create CD8+ Tcell phenotype ☐
  - Assign functions ☐
  - Compile and test ☐

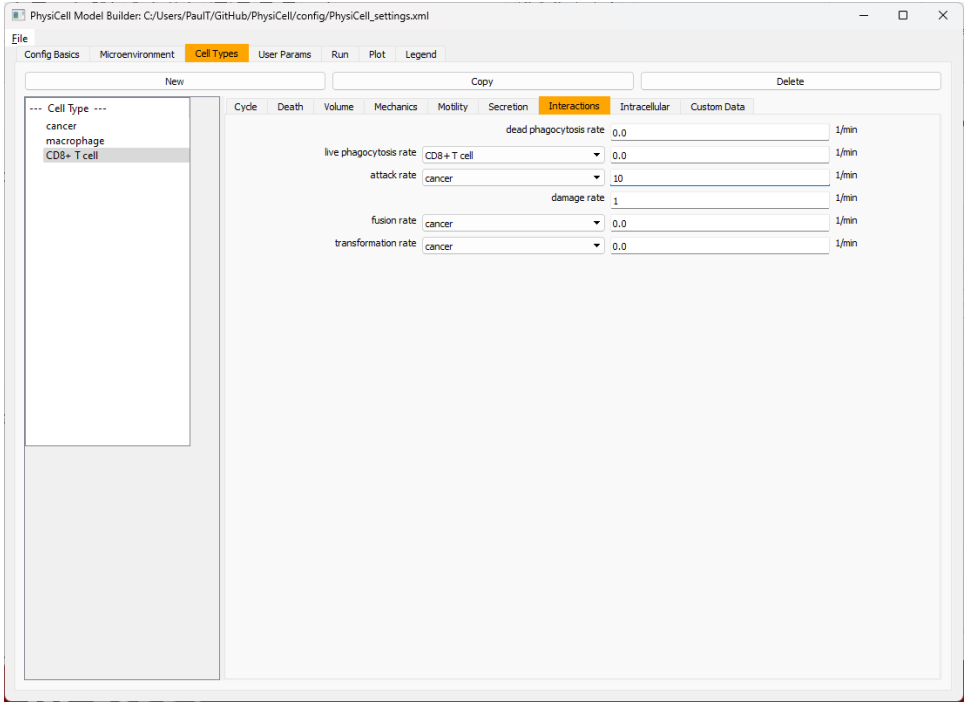
Create CD8+ T cells: motility & secretion

- Click the **cell types** tab
- select **macrophage** and copy
  - rename to **CD8+ T cell**
  - go to **motility**
    - Set **migration bias** to **0.5**
    - Go to **chemotaxis**
      - » Choose **pro-inflammatory** from the drop-down
  - go to **secretion**
    - Set **debris** uptake/secretion to 0
    - Set **oxygen** uptake to 10
    - Set **pro-inflammatory** uptake to 1
      - » Ligand-receptor binding uses ligand
    - Set **anti-inflammatory** uptake to 1
      - » Ligand-receptor binding uses ligand



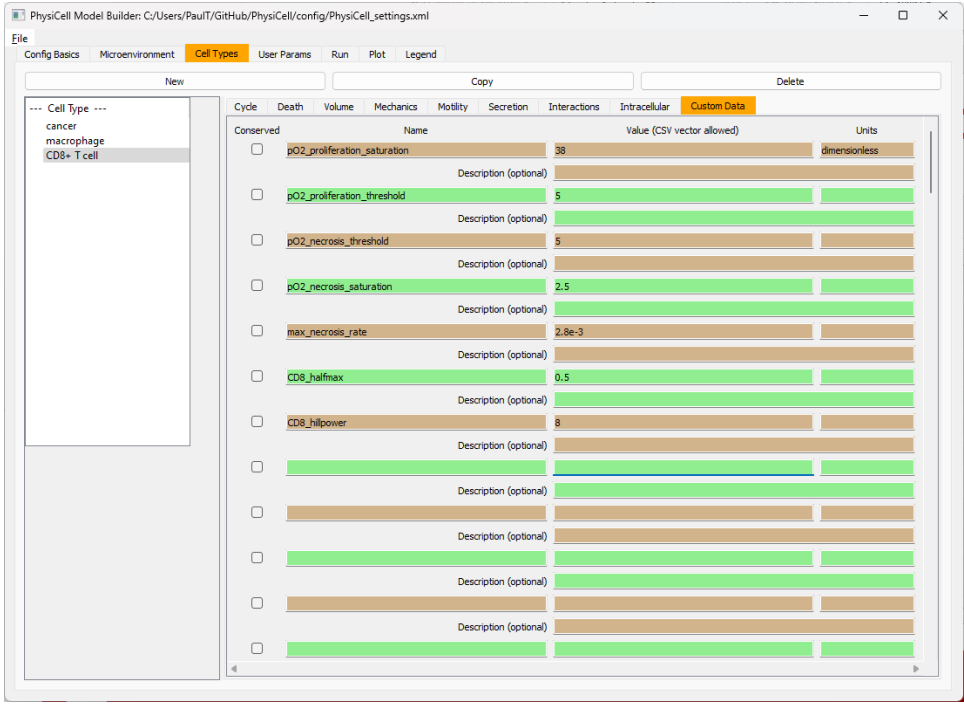
Create CD8+ T cells: interactions

- select **interactions** tab
  - Set all **phagocytosis** rates to 0
  - Go to **attack rate**
    - ♦ Choose **cancer** from the drop-down
    - ♦ Set **attack rate** to 10
      - » Attacks every 0.1 min (thus at every mechanics step)



Create CD8+ T cells: custom data

- Go to **custom data** tab
  - Add **CD8_halfmax** = **0.5**
  - Add **CD8_hillpower** = **8**



Checklist

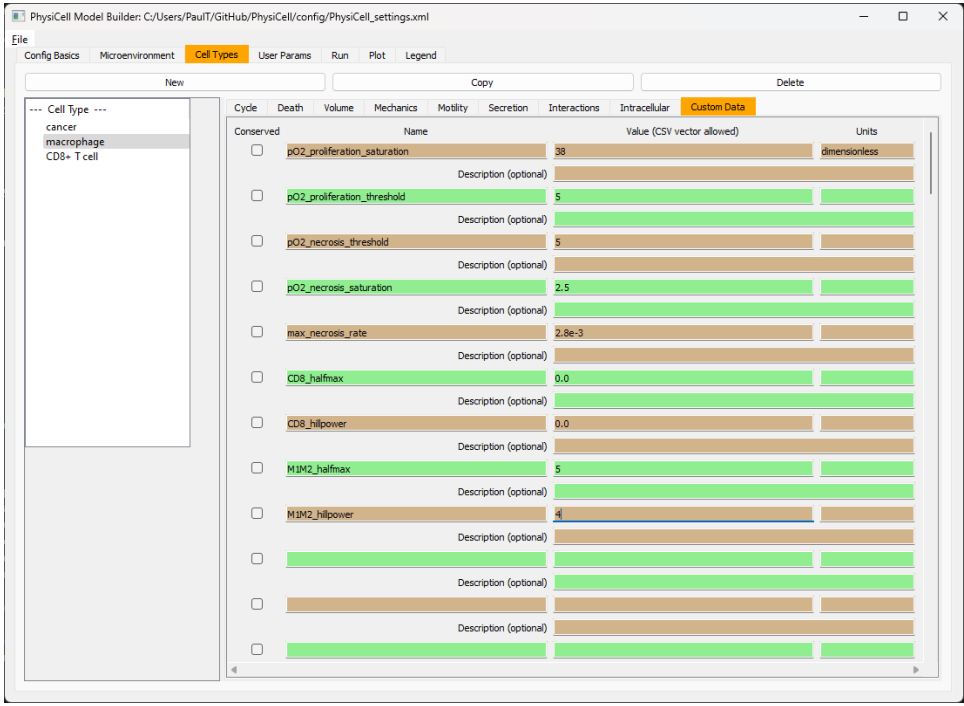
- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☐
  - Update cancer cells ☐
- Refinement (in C++):
  - Update cancer cell phenotype ☐
  - Create macrophage phenotype ☐
  - Create CD8+ Tcell phenotype ☐
  - Assign functions ☐
  - Compile and test ☐

Update macrophages: secretion

- Select **macrophage**
- Go to **secretion**
  - Choose **pro-inflammatory** in the drop-down
    - set **secretion rate** to 10
    - set **target** to 1
  - Choose **anti-inflammatory** in the drop-down
    - set **secretion rate** to 10
    - set **target** to 1

Update macrophages: custom data

- Go to **custom data**
  - Add M1M2_halfmax = 5
  - Add M1M2_hillpower = 4

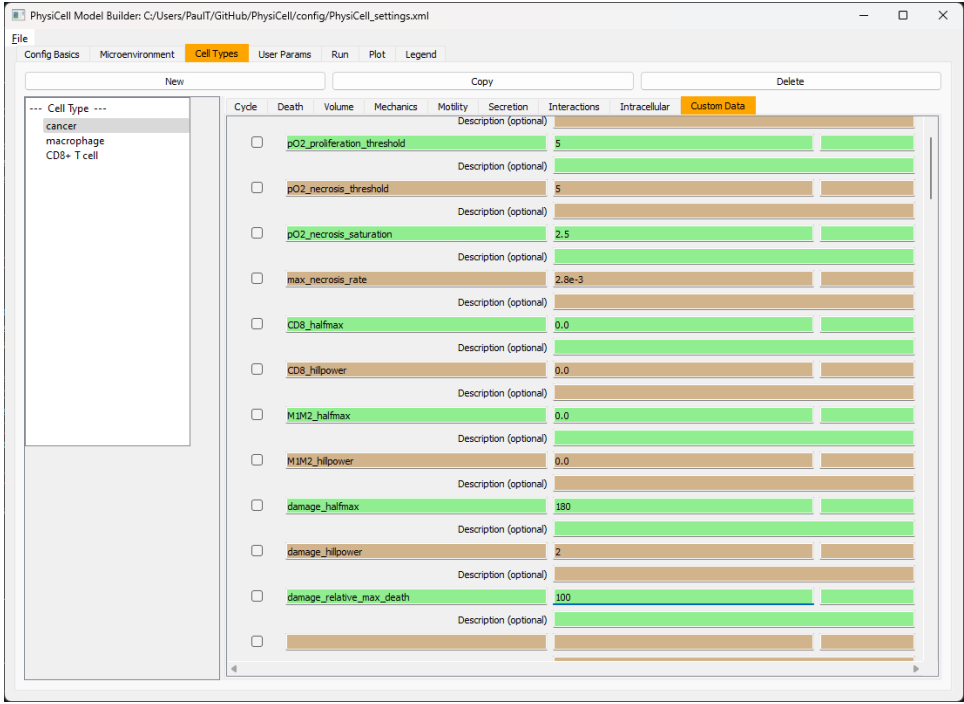


Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☑
  - Update cancer cells ☐
- Refinement (in C++):
  - Update cancer cell phenotype ☐
  - Create macrophage phenotype ☐
  - Create CD8+ Tcell phenotype ☐
  - Assign functions ☐
  - Compile and test ☐

Update cancer: custom data

- Select **cancer** as the cell type
- Go to **custom data**
  - Add …



Let's add two parameters

- Parameters for CD8+ T cell placement
  - go to **user params**
    - add **number_of_CD8_Tcells** of type **int** with value **50**
    - add **CD8_Tcell_distance** of type **double** with value **375**

Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☑
  - Update cancer cells ☑
- Refinement (in C++):
  - Update cancer cell phenotype ☐
  - Create macrophage phenotype ☐
  - Create CD8+ T cell phenotype ☐
  - Assign functions ☐
  - Compile and test ☐

Unzip Session07_checkpoint1.zip in ./PhysiCell to get this code.

Declare custom functions

- In **`./custom_modules/custom.h`**, declare:

```
void macrophage_phenotype( Cell* pCell, Phenotype& p, double dt);
void CD8_Tcell_phenotype( Cell* pCell, Phenotype& p, double dt);
```

Update cancer phenotype

```
void cancer_phenotype( Cell* pCell, Phenotype& p, double dt)
{
   // ... from last time

   // set apoptosis rate based on damage

   double damage = get_single_signal( pCell , "damage");
   rate0 = get_single_base_behavior( pCell , "apoptosis");
   double halfmax = get_single_behavior( pCell , "custom:damage_halfmax"); // 180
   double hillpower = get_single_behavior( pCell , "custom:damage_hillpower"); // 2
   rateMax = get_single_behavior( pCell , "custom:damage_relative_max_death") * rate0;
   double hill = Hill_response_function( damage , halfmax , hillpower );
   rate = rate0 + (rateMax-rate0)*hill;
   set_single_behavior( pCell , "apoptosis" , rate) ;

   return;
}
```

Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☑
  - Update cancer cells ☑
- Refinement (in C++):
  - Update cancer cell phenotype ☑
  - Create macrophage phenotype ☐
  - Create CD8+ T cell phenotype ☐
  - Assign functions ☐
  - Compile and test ☐

macrophage phenotype

```
void macrophage_phenotype( Cell* pCell, Phenotype& p, double dt)
{
   // secrete anti-inflammatory in low O2

   double o2 = get_single_signal( pCell, "oxygen");
   double halfmax = get_single_behavior( pCell , "custom:M1M2_halfmax"); // 5
   double hillpower = get_single_behavior( pCell , "custom:M1M2_hillpower"); // 4
   double hill = Hill_response_function( o2 , halfmax, hillpower );
   double rate = get_single_base_behavior( pCell, "anti-inflammatory secretion")*(1-hill);
   set_single_behavior( pCell , "anti-inflammatory secretion" , rate);

   // secrete pro-inflammatory except in low O2

   rate = get_single_base_behavior( pCell, "pro-inflammatory secretion")*hill;
   set_single_behavior( pCell , "pro-inflammatory secretion" , rate);

   return;
}
```

Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☑
  - Update cancer cells ☑
- Refinement (in C++):
  - Update cancer cell phenotype ☑
  - Create macrophage phenotype ☑
  - Create CD8+ T cell phenotype ☐
  - Assign functions ☐
  - Compile and test ☐

CD8+ T cell phenotype

```
void CD8_Tcell_phenotype( Cell* pCell, Phenotype& p, double dt)
{
  // anti-inflammatory reduces motiltiy

  double anti = get_single_signal(pCell, "anti-inflammatory");
  double halfmax = get_single_behavior( pCell , "custom:CD8_halfmax"); // 0.5
  double hillpower = get_single_behavior( pCell , "custom:CD8_hillpower"); // 8
  double hill = Hill_response_function( anti , halfmax , hillpower );

  double param = get_single_base_behavior( pCell, "migration speed") * (1-hill);
  set_single_behavior( pCell , "migration speed" , param );

  // anti-inflammatory reduces cell killing

  param = get_single_base_behavior( pCell, "attack cancer") * (1-hill);
  set_single_behavior( pCell , "attack cancer" , param );

  return;
}
```

Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☑
  - Update cancer cells ☑
- Refinement (in C++):
  - Update cancer cell phenotype ☑
  - Create macrophage phenotype ☑
  - Create CD8+ T cell phenotype ☑
  - Assign functions ☐
  - Compile and test ☐

Assign the functions

```
// in create_cell_types():

  // ...

  Cell_Definition* pCD = find_cell_definition( "cancer" );
  pCD->functions.update_phenotype = cancer_phenotype;

  pCD = find_cell_definition( "macrophage" );
  pCD->functions.update_phenotype = macrophage_phenotype;

  pCD = find_cell_definition( "CD8+ T cell" );
  pCD->functions.update_phenotype = CD8_Tcell_phenotype;

  /*
     This builds the map of cell definitions and summarizes the setup.
  */

  // ...
```

Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☑
  - Update cancer cells ☑
- Refinement (in C++):
  - Update cancer cell phenotype ☑
  - Create macrophage phenotype ☑
  - Create CD8+ T cell phenotype ☑
  - Assign functions ☑
  - Compile and test ☐

Place CD8+ T cells (almost forgot!)

```
// in setup_tissue ():

  // place CD8+ T cells

  pCD = find_cell_definition( "CD8+ T cell" );

  std::cout << "Placing cells of type " << pCD->name << " ... " << std::endl;
  for( int k=0 ; k < parameters.ints( "number_of_CD8_Tcells" ); k++ )
  {
    std::vector<double> position = UniformOnUnitCircle();
    position *= parameters.doubles("CD8_Tcell_distance");

    pC = create_cell( *pCD );
    pC->assign_position( position );
  }

  // load cells from your CSV file (if enabled)
  load_cells_from_pugixml();

  // ...
```

Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☑
  - Update cancer cells ☑
- Refinement (in C++):
  - Update cancer cell phenotype ☑
  - Create macrophage phenotype ☑
  - Create CD8+ T cell phenotype ☑
  - Assign functions ☑
  - Compile and test ☐

Let's update the coloring function

- Shade **macrophages** from red (M1-like) to white (M2-like)
  - Base coloring on the anti-inflammatory secretion rate.

- Color **CD8+ T** cells dark green

Updated coloring function (1)

```
std::vector<std::string> custom_coloring_function( Cell* pCell )
{
  // ...

  // if live CD8+ T cell, color dark green

  if( pCell->type_name == "CD8+ T cell" && dead == false )
  {
    // get relative birth rate
    char szColor [1024] = "rgb(0,128,0)";

    // modify output
    output[0] = szColor;
    output[2] = szColor;
    output[3] = szColor;
  }

  // ...
}
```

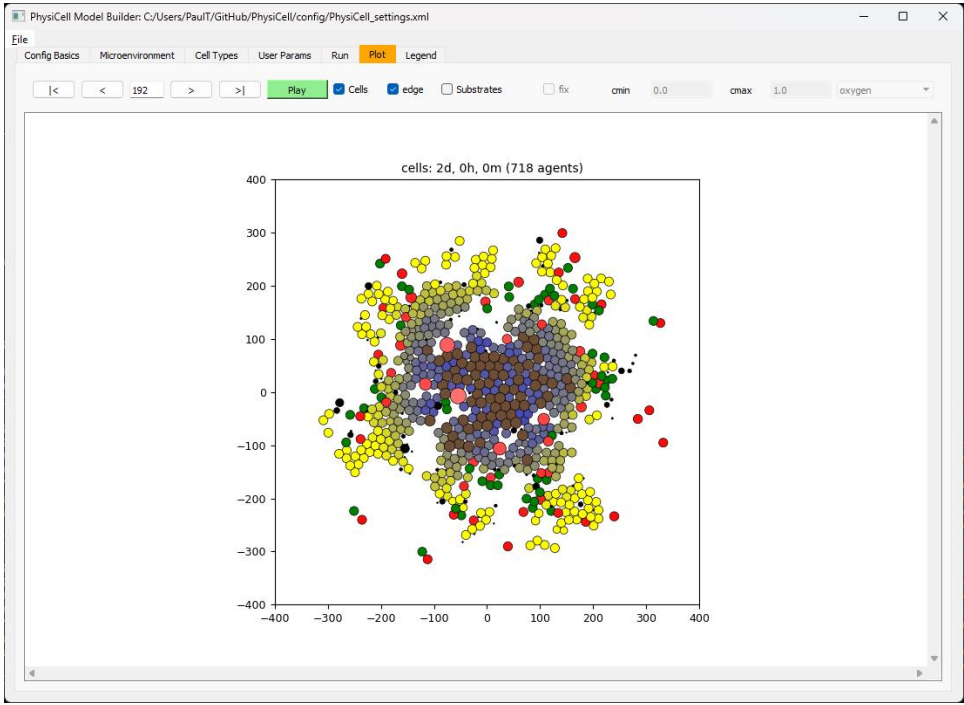
Updated coloring function (2)

```
std::vector<std::string> custom_coloring_function( Cell* pCell )
{
   // ...

   // live macrophage cells: shade by anti-inflammatory state

   if( pCell->type_name == "macrophage" && dead == false )
   {
      // get relative secretion rate
      double s = 1 * get_single_behavior( pCell, "anti-inflammatory secretion" )
         / get_single_base_behavior( pCell, "anti-inflammatory secretion" );
      if( s > 1 )
      { s = 1; }

      // make color
      int color = (int) round( 255.0 * s );
      char szColor [1024];
      // interpolate from red to white
      sprintf( szColor, "rgb(%u,%u,%u)",255,color,color );

      // modify output
      output[0] = szColor;
      output[2] = szColor;
      output[3] = szColor;
   }

   // ...
}
```

Rebuild and test the model

- In your terminal window, recompile:
  - **make**
- Go to the **run** tab and click **run**
- Go to the **plot** tab
  - click **play** to animate
- View the **legend** tab to see the cell colors
- Expected behavior:
  - Tumor cell growth faster near outer edge
    - Bright yellow = rapidly proliferating
  - Macrophages wander towards dead cells
    - necrotic core (brown), sporadic apoptosis (black)
    - phagocytosis of dead cells *if they can reach them*
    - more M2-like (white) in hypoxic regions
  - CD8+ T cells attack and kill cancer cells

cancer
macrophage



Unzip Session07_checkpoint2.zip in ./PhysiCell to get this code.

Checklist

- Plan ☑
- Build iteratively (in model builder):
  - Add (new) diffusing substrates ☑
  - Add CD8+ T cells ☑
  - Update macrophages ☑
  - Update cancer cells ☑
- Refinement (in C++):
  - Update cancer cell phenotype ☑
  - Create macrophage phenotype ☑
  - Create CD8+ T cell phenotype ☑
  - Assign functions ☑
  - Compile and test ☑

New problem: zombies!

- **farmers**:
  - release food
  - chemotax away from food (create where it's most needed)

- **prey**:
  - consume food
  - reproduce, low background death rate
    - birth rate proportional to food
  - release quorum factor
  - chemotax towards food, towards quorum factor, away from zombies
    - speed proportional to food
  - (zombie-induced) damage causes transformation into zombie
  - can lightly counter-attack zombies

- **zombies:**
  - attack prey
  - eat the dead
  - chemotax towards
  - fuse into megazombies
    - higher damage rate, lower speed
  - (prey-induced) damage causes death

Full modeling workflow

Suitable for creating a new PhysiCell model with custom C++ to drive dynamical phenotype changes

- Plan the model
- Populate a project
- Edit configuration Model Builder GUI
  - Edit domain
  - Edit microenvironment
  - Edit cell definitions
  - **Add custom variables**
  - **Add custom parameters**
- **Edit custom modules:**
  - **Declare functions in custom.h**
  - **Implement functions in custom.cpp**
  - **Assign functions to cell definitions**
- **Edit initial cell placement**
- **Edit cell coloring function**
- Build
- Run
- View results

Checklist

- ❑ Plan
- ❑ Populate and compile template project
- ❑ Edit domain
- ❑ Edit microenvironment
  - ❑ food
  - ❑ quorum
  - ❑ zombie
- ❑ Create cell definitions
  - ❑ farmer
  - ❑ prey
  - ❑ zombie
- ❑ Add custom variables
- ☑ ~~Add custom parameters~~
- ❑ Declare custom functions
- ❑ Implement custom functions
  - ❑ prey_phenotype
  - ❑ zombie_phenotype
- ❑ Assign custom functions
- ☑ ~~Initial cell placement~~
- ☑ ~~Create and assign coloring~~
- ❑ Compile and test!

prey mathematics

- **proliferation:** simple linear model (f = food)

$$b = b_0 \cdot f$$

  - b_0 1e-3

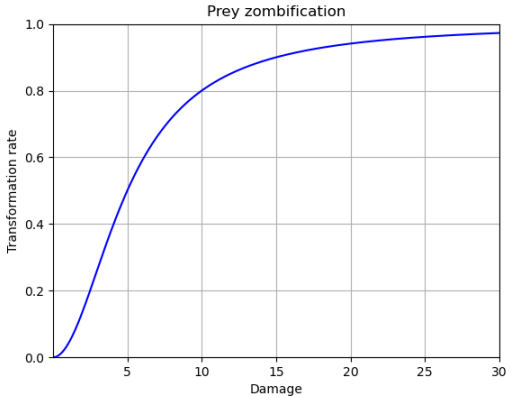
- **speed:** simple linear model (f = food)

$$s = s_0 \cdot f$$

- **zombification:** Hill response function (D = damage)

$$r_{\text{zombie}} = r_0 \cdot H(D)$$

  - r_0 0.01
  - Half-max 5
  - Hill-power 2



zombie mathematics

- **speed:** Hill response model (n = number of nuclei)

$$s = s_0 \cdot \left(1 - H(n)\right)$$

  - half-max 2
  - Hill power 2

- **attack:** linear response model (n = number of nuclei)

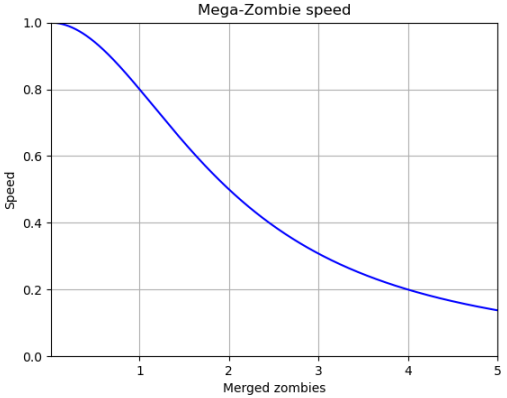
$$r_{\text{damage}} = r_0 \cdot n$$

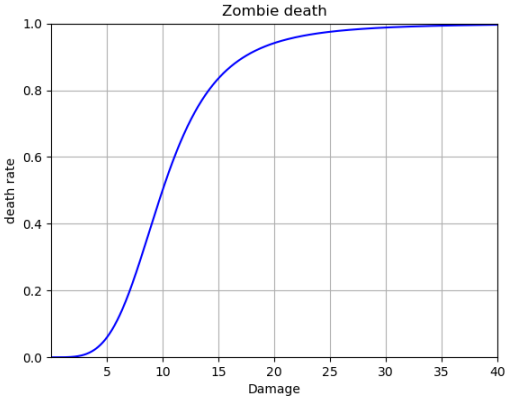
  - r_0 1

- **death:** Hill response model (D = damage)

$$d = d_M \cdot H(D)$$

  - d_M 0.01
  - half-max 20
  - Hill power 4





Custom variables

- prey_tr_halfmax
- prey_tr_hillpower
- zombie_speed_halfmax
- zombie_speed_hillpower
- zombie_damage_halfmax
- zombie_damage_hillpower

Checklist

- ☑ Plan
- ❑ Populate and compile template project
- ❑ Edit domain
- ❑ Edit microenvironment
  - ❑ food
  - ❑ quorum
  - ❑ zombie
- ❑ Create cell definitions
  - ❑ farmer
  - ❑ prey
  - ❑ zombie
- ❑ Add custom variables
- ❑ ~~Add custom parameters~~
- ❑ Declare custom functions
- ❑ Implement custom functions
  - ❑ prey_phenotype
  - ❑ zombie_phenotype
- ❑ Assign custom functions
- ❑ ~~Initial cell placement~~
- ❑ ~~Create and assign coloring~~
- ❑ Compile and test!

Reset and get ready

- return to blank slate
  - make reset
- clean out data
  - make data-cleanup
- populate and build template project
  - make template
  - make

Checklist

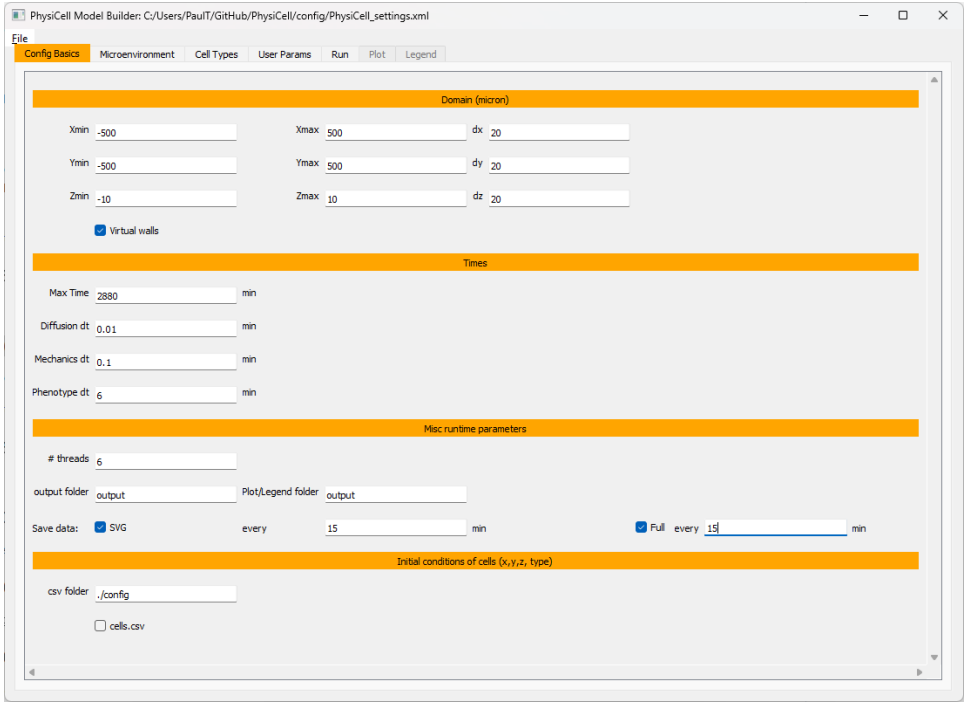
- ☑ Plan
- ☑ Populate and compile template project
- ❑ Edit domain
- ❑ Edit microenvironment
  - ❑ food
  - ❑ quorum
  - ❑ zombie
- ❑ Create cell definitions
  - ❑ farmer
  - ❑ prey
  - ❑ zombie
- ❑ Add custom variables
- ~~❑ Add custom parameters~~
- ❑ Declare custom functions
- ❑ Implement custom functions
  - ❑ prey_phenotype
  - ❑ zombie_phenotype
- ❑ Assign custom functions
- ~~❑ Initial cell placement~~
- ~~❑ Create and assign coloring~~
- ❑ Compile and test!

Edit domain

- open config/PhysiCell_settings.xml

- Go to **config basics** tab
  - We'll leave the spatial domain as-is
  - Set max time to 2880 min (2 days)
  - Output SVG every 15 minutes
  - Output full data every 15 minutes

- save

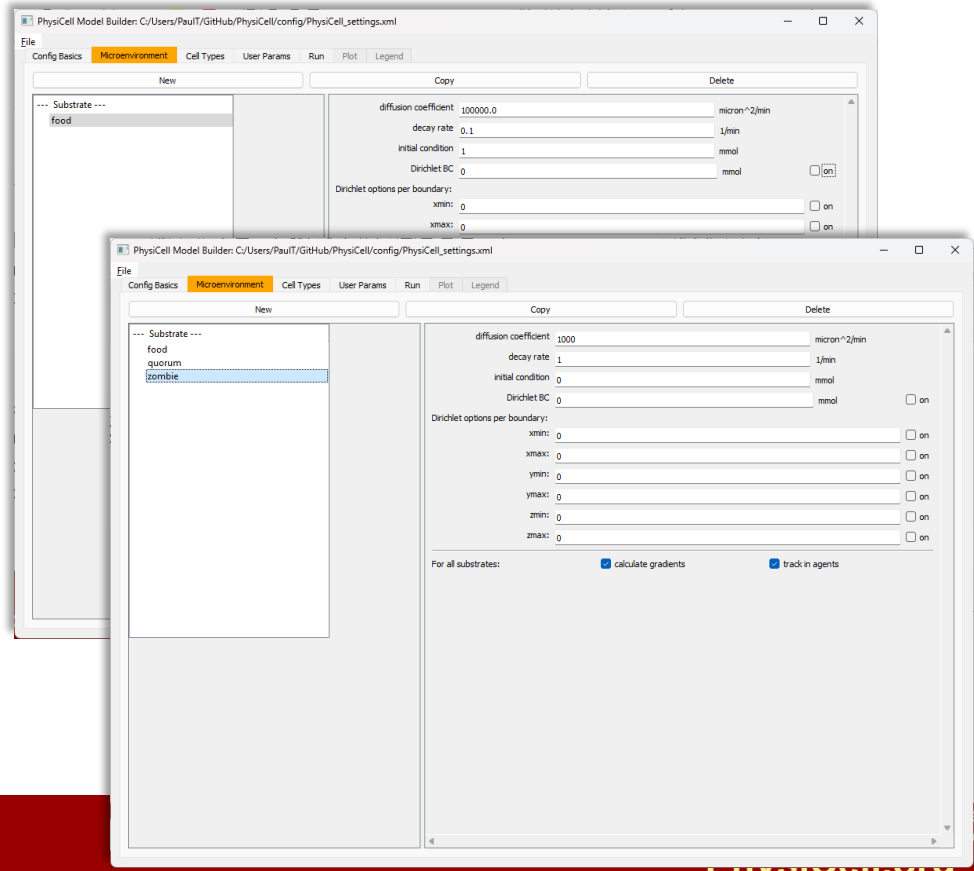


Checklist

- ☑ Plan
- ☑ Populate and compile template project
- ☑ Edit domain
- ❑ Edit microenvironment
  - ❑ food
  - ❑ quorum
  - ❑ zombie
- ❑ Create cell definitions
  - ❑ farmer
  - ❑ prey
  - ❑ zombie
- ❑ Add custom variables
- ~~❑ Add custom parameters~~
- ❑ Declare custom functions
- ❑ Implement custom functions
  - ❑ prey_phenotype
  - ❑ zombie_phenotype
- ❑ Assign custom functions
- ~~❑ Initial cell placement~~
- ~~❑ Create and assign coloring~~
- ❑ Compile and test!

Edit microenvironment

- Go to **microenvironment tab**
  - double-click **substrate**
    - rename to **food**
    - set decay to 0.01
    - set **initial condition** to 1
    - disable Dirichlet boundary
  - select **food** and copy
    - rename to **quorum**
    - set **diffusion** to 1000
    - set **decay** to 1
    - set **initial condition** to 0
  - select **quorum** and copy
    - rename to **zombie**
  - Save

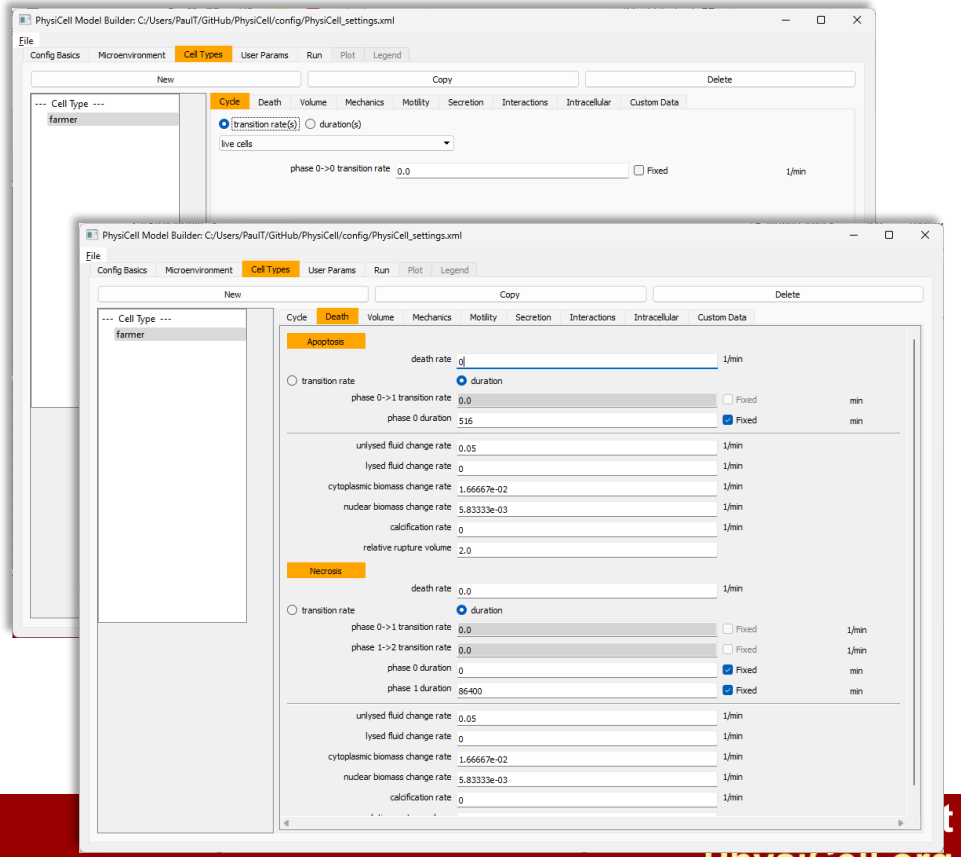


Checklist

- ☑ Plan
- ☑ Populate and compile template project
- ☑ Edit domain
- ☑ Edit microenvironment
  - ☑ food
  - ☑ quorum
  - ☑ zombie
- ❑ Create cell definitions
  - ❑ farmer
  - ❑ prey
  - ❑ zombie
- ❑ Add custom variables
- ❑ ~~Add custom parameters~~
- ❑ Declare custom functions
- ❑ Implement custom functions
  - ❑ prey_phenotype
  - ❑ zombie_phenotype
- ❑ Assign custom functions
- ❑ ~~Initial cell placement~~
- ❑ ~~Create and assign coloring~~
- ❑ Compile and test!

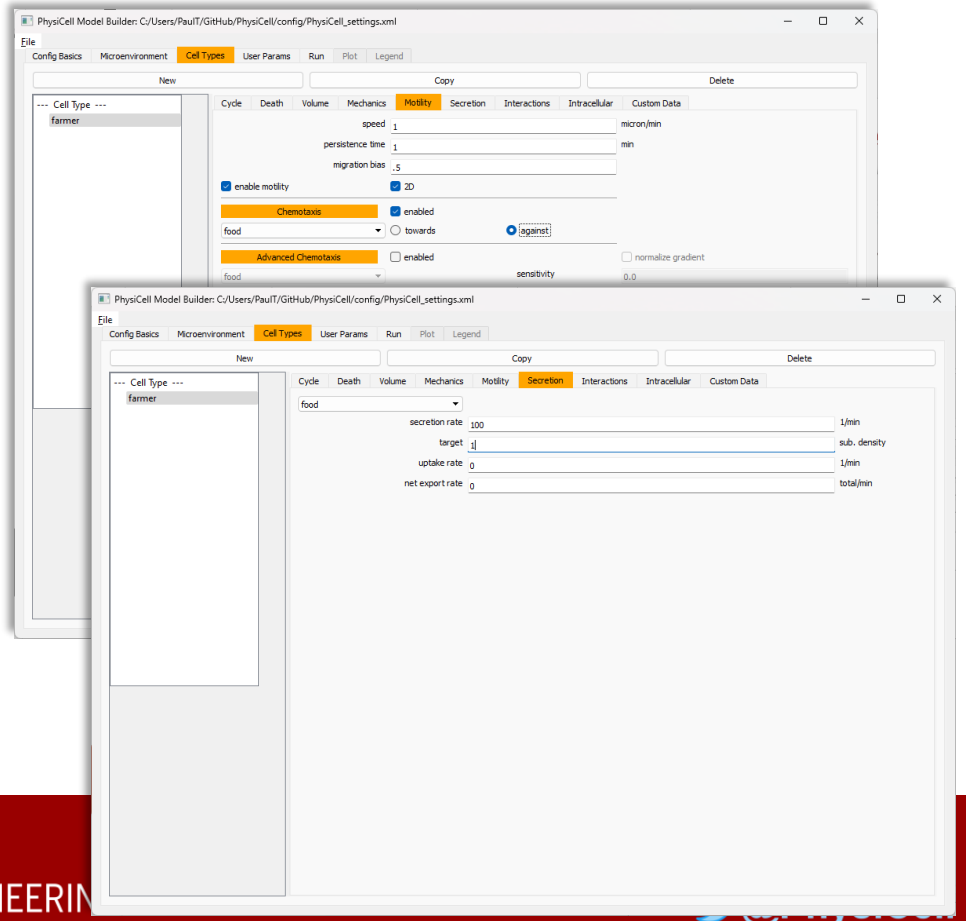
farmers: proliferation and death

- Go to **cell definitions**
- double-click **default**
  - rename to **farmer**
  - go to **cycle**
    - choose the **live** cells model with transitions
    - keep cycle rate at 0
  - go to **death**
    - set the **apoptosis rate** to 0



farmers: mechanics, motility, secretion

- go to **mechanics**
  - set **cell-cell adhesion** to 0
- go to **motility**
  - leave speed, bias, persistence
  - check **enabled**
  - go to **chemotaxis**
    - choose **food** from drop down, with **against**
    - set **enable**
- go to **secretion**
  - choose **food** in the drop-down
    - set **secretion rate** to 100
    - set **target** to 1
- save



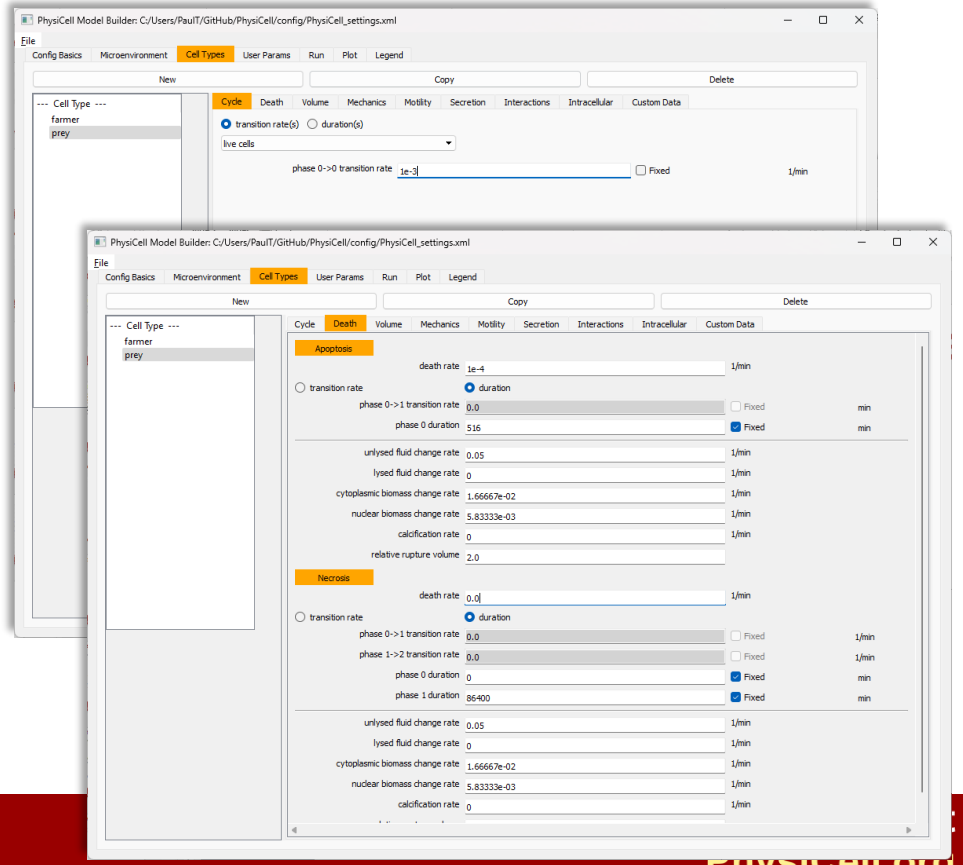
Checklist

- [x] Plan
- [x] Populate and compile template project
- [x] Edit domain
- [x] Edit microenvironment
  - [x] food
  - [x] quorum
  - [x] zombie
- [ ] Create cell definitions
  - [x] farmer
  - [ ] prey
  - [ ] zombie
- [ ] Add custom variables
- [~~x~~] ~~Add custom parameters~~
- [ ] Declare custom functions
- [ ] Implement custom functions
  - [ ] prey_phenotype
  - [ ] zombie_phenotype
- [ ] Assign custom functions
- [~~x~~] ~~Initial cell placement~~
- [~~x~~] ~~Create and assign coloring~~
- [ ] Compile and test!

prey: proliferation and death

- copy **farmer** and copy
  - rename to **prey**
  - go to **cycle**
    - choose the **live** cells model with transitions
    - set **transition rate** to 1e-3
  - go to **death**
    - set the **apoptosis rate** to 1e-4

- save

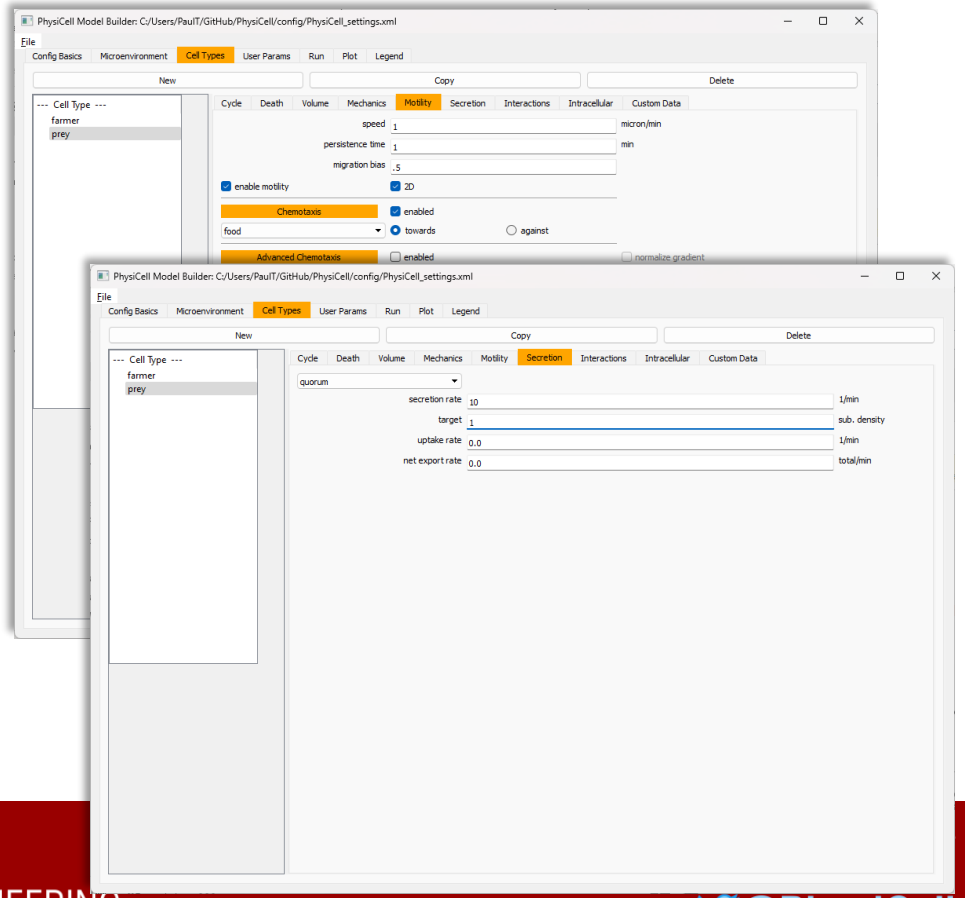


prey: motility and secretion

- go to **motility**
  - change taxis towards **food**

- go to **secretion**
  - choose **food** from the drop-down
    - ♦ set **secretion rate** to 0
    - ♦ set **uptake rate** to 10
  - choose **quorum factor**
    - ♦ set **secretion rate** to 10
    - ♦ set **target** to 1

- save



Notes

- We'll redo motility once we add zombies
- We'll do all the interactions at once

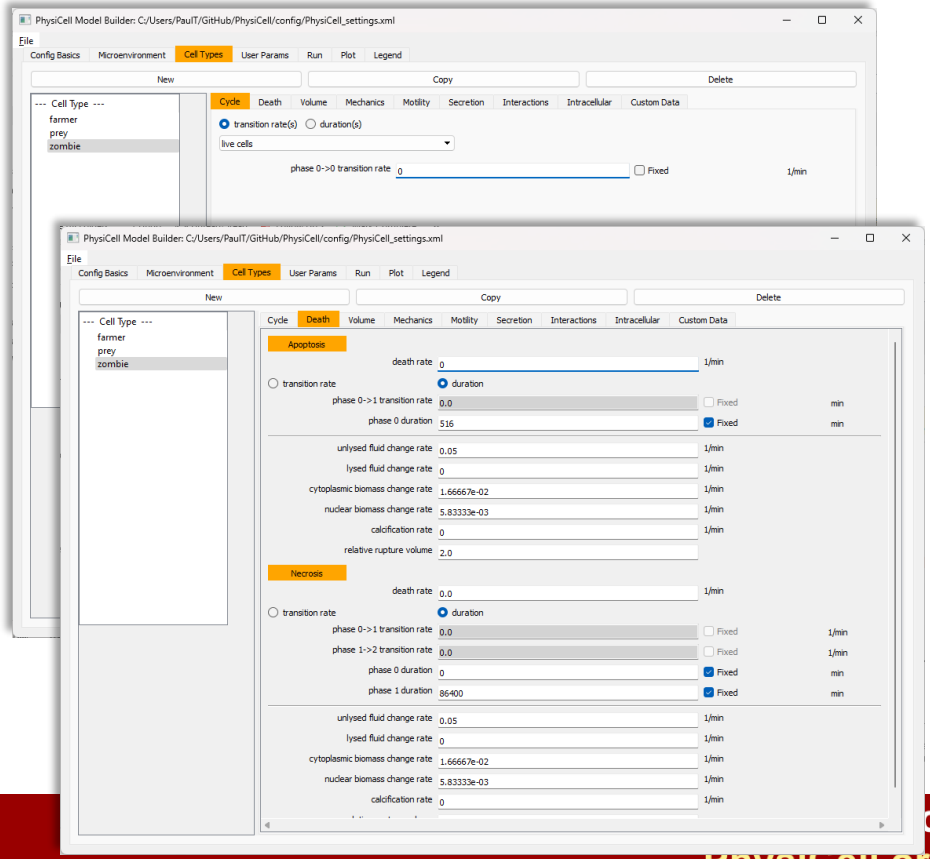
Checklist

- [x] Plan
- [x] Populate and compile template project
- [x] Edit domain
- [x] Edit microenvironment
  - [x] food
  - [x] quorum
  - [x] zombie
- [ ] Create cell definitions
  - [x] farmer
  - [x] prey
  - [ ] zombie
- [ ] Add custom variables
- [~] ~~Add custom parameters~~
- [ ] Declare custom functions
- [ ] Implement custom functions
  - [ ] prey_phenotype
  - [ ] zombie_phenotype
- [ ] Assign custom functions
- [~] ~~Initial cell placement~~
- [~] ~~Create and assign coloring~~
- [ ] Compile and test!

zombie: proliferation and death

- click **prey** and copy
  - rename to **zombie**
  - go to **cycle**
    - ♦ choose the **live** cells model with transitions
    - ♦ set **transition rate** to 0
  - go to **death**
    - ♦ set the **apoptosis rate** to 0

- save

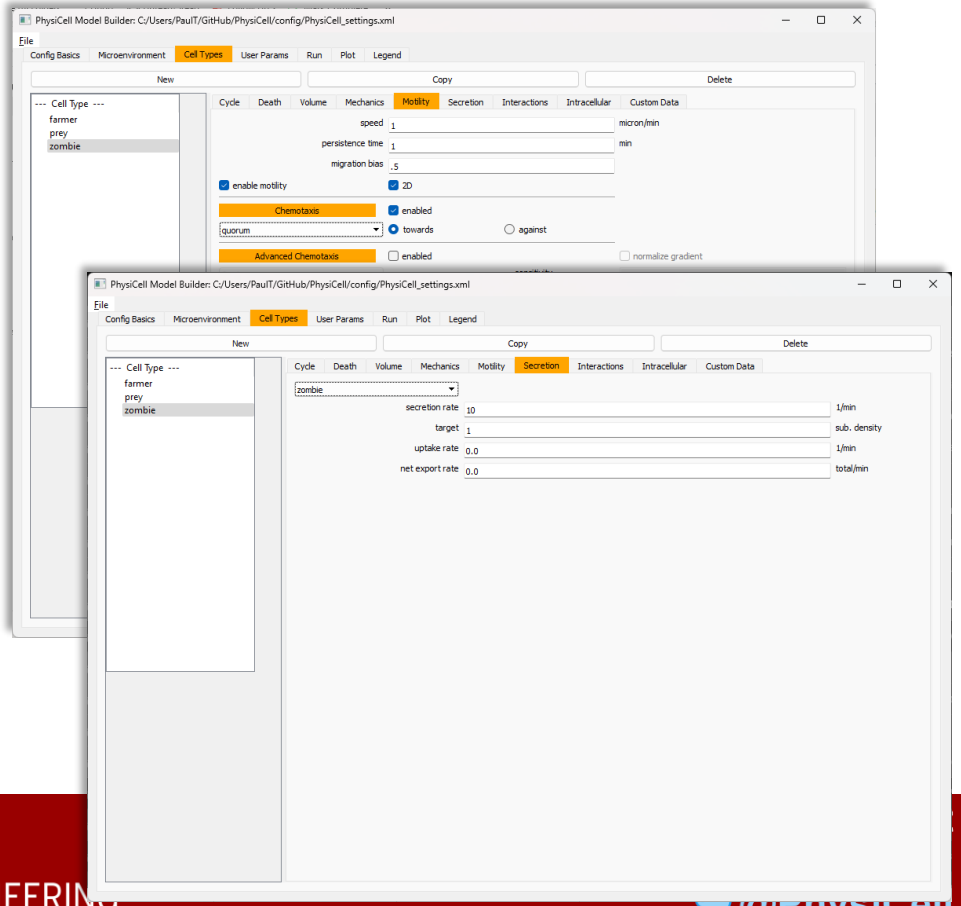


zombie: motility and secretion

- go to **motility**
  - go to **chemotaxis**
    - ♦ choose **quorum** from the drop-down

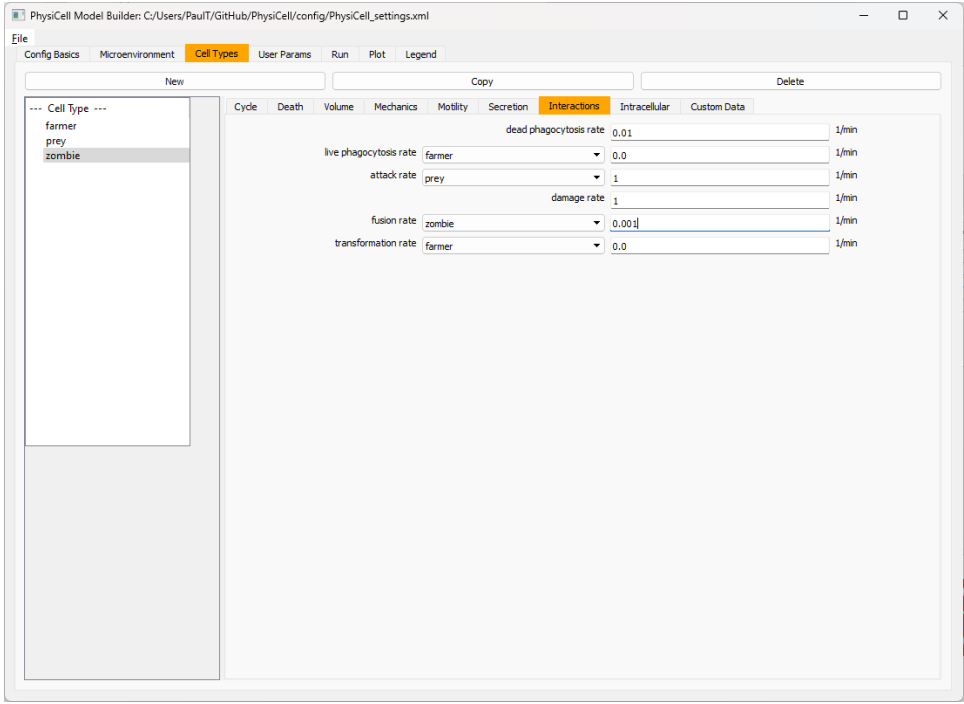
- go to **secretion**
  - ♦ choose **food** from the drop-down
    - » set **uptake rate** to 0
  - ♦ the quorum from the drop-down
    - » set **secretion rate** to 0
  - ♦ choose **zombie** from the drop-down
    - » set **secretion rate** to 10
    - » set **target** to 1

- save



zombie interactions

- go to **interactions**
  - set **dead phagocytosis** to 0.01
  - go to **attack rate**
    - ♦ choose **prey** from the drop-down
    - ♦ set the **attack rate** to 1
  - go to **fusion rate**
    - ♦ choose **zombie** from the drop-down
    - ♦ set the **fusion rate** to 0.001



Checklist

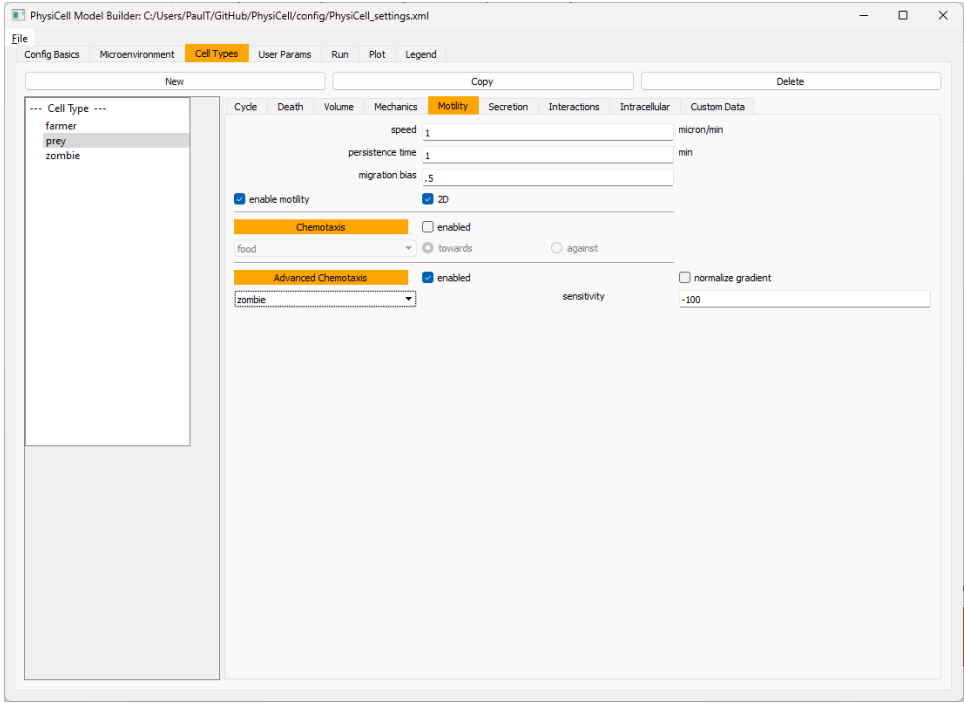
- [x] Plan
- [x] Populate and compile template project
- [x] Edit domain
- [x] Edit microenvironment
  - [x] food
  - [x] quorum
  - [x] zombie
- [ ] Create cell definitions
  - [x] farmer
  - [x] prey
  - [x] zombie
- [ ] Add custom variables
- [~] ~~Add custom parameters~~
- [ ] Declare custom functions
- [ ] Implement custom functions
  - [ ] prey_phenotype
  - [ ] zombie_phenotype
- [ ] Assign custom functions
- [~] ~~Initial cell placement~~
- [~] ~~Create and assign coloring~~
- [ ] Compile and test!

prey: advanced chemotaxis

- Need to:
  - move towards food (high preference)
  - move towards prey (med pref)
  - move away from zombies (high pref)

- choose **prey** go to **motility**
  - go to **advanced chemotaxis**
    - ♦ set **enabled**
    - ♦ choose **food** from the drop-down
      - » set **sensitivity** to 10
    - ♦ choose **quorum** from the drop-down
      - » set **sensitivity** to 1
    - ♦ choose **zombie** from the drop-down
      - » set **sensitivity** to -100

- save

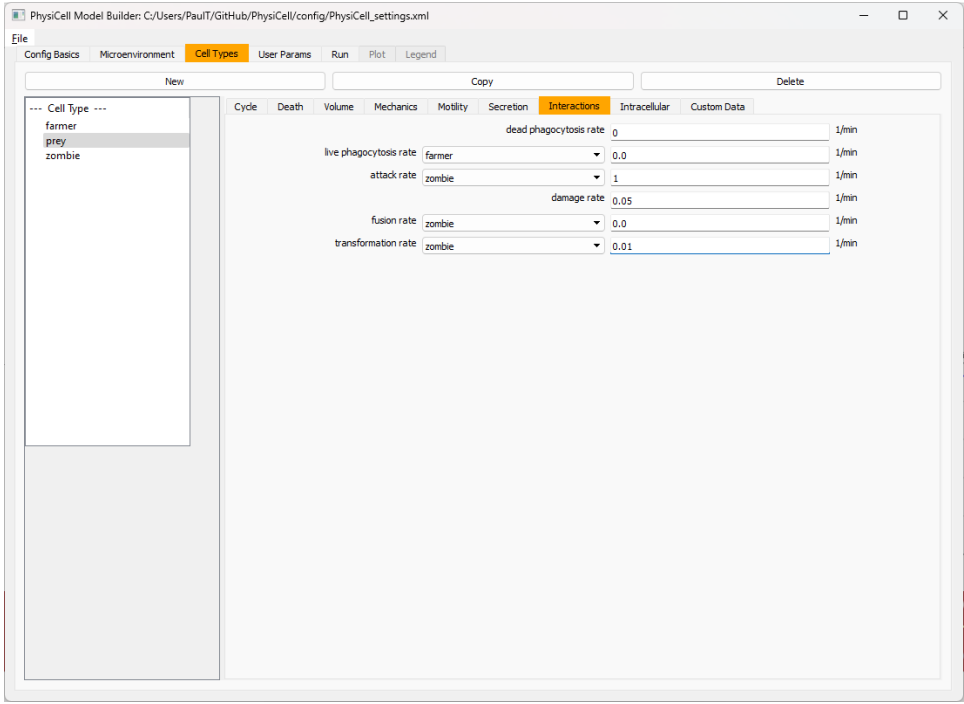


prey: finalize interactions

- Need to:
  - let prey counter-attack (weakly)

- choose **prey** go to **interactions**
  - go to **attack**
    - ♦ choose **zombie** in the drop-down
      - » set **attack rate** to 1
      - » set **damage rate** to 0.05
    - ♦ go to **transformations**
      - » choose **zombie** in the drop-down
      - » set **rate** to 0.01

- save

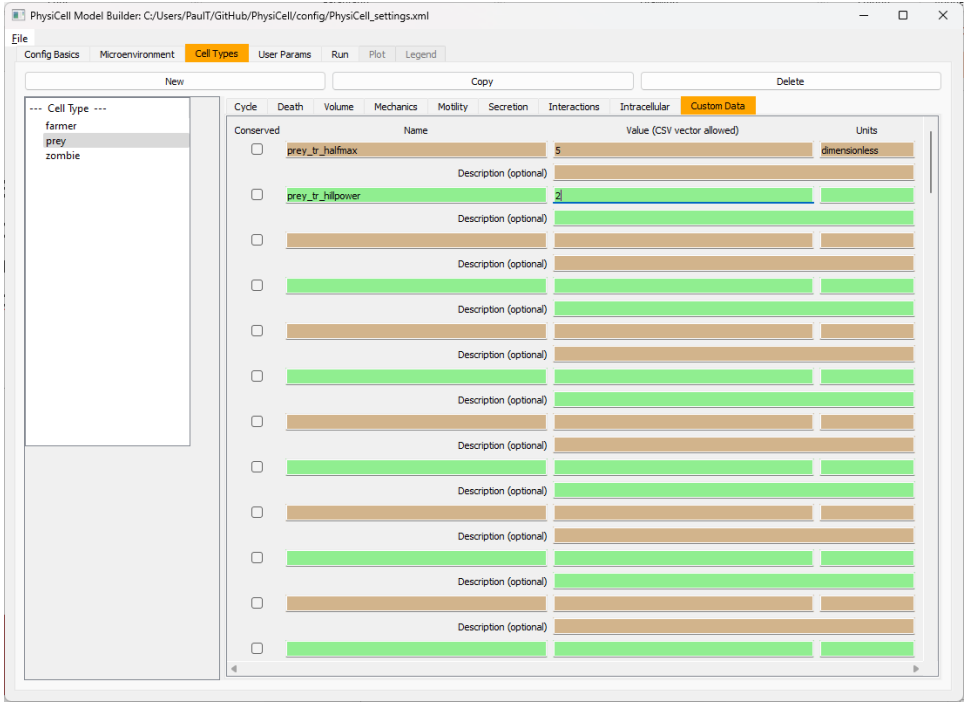


Checklist

- [x] Plan
- [x] Populate and compile template project
- [x] Edit domain
- [x] Edit microenvironment
  - [x] food
  - [x] quorum
  - [x] zombie
- [ ] Create cell definitions
  - [x] farmer
  - [x] prey
  - [x] zombie
- [ ] Add custom variables
- [~~x~~] ~~Add custom parameters~~
- [ ] Declare custom functions
- [ ] Implement custom functions
  - [ ] prey_phenotype
  - [ ] zombie_phenotype
- [ ] Assign custom functions
- [~~x~~] ~~Initial cell placement~~
- [~~x~~] ~~Create and assign coloring~~
- [ ] Compile and test!

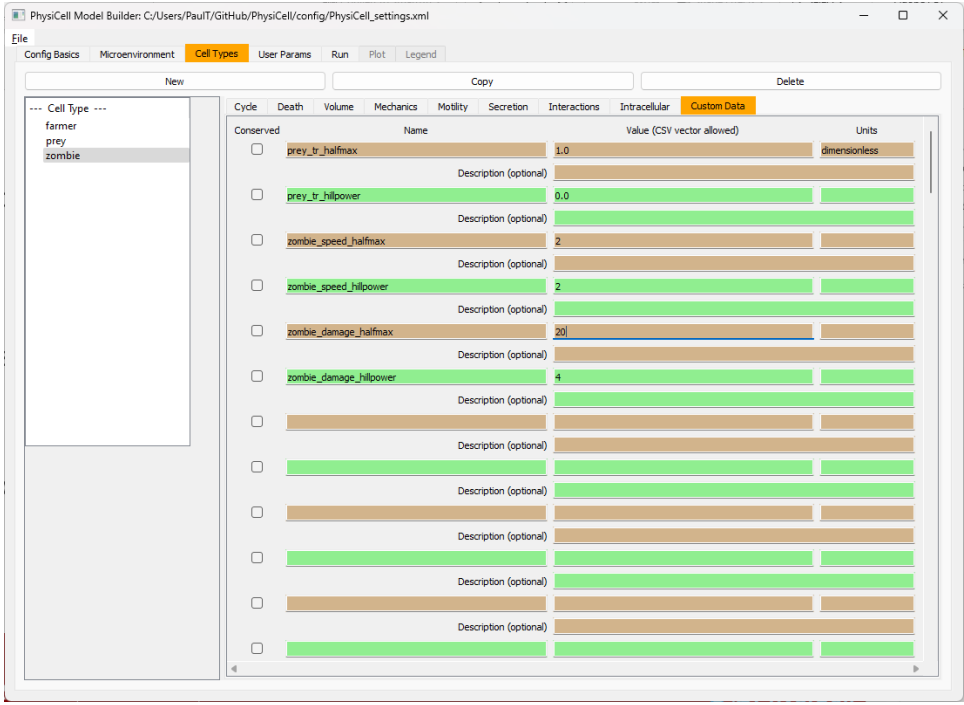
prey: custom variables

- choose **prey**
  - go to **custom data**
    - ♦ rename **sample** to **prey_tr_halfmax** = 5
    - ♦ add **prey_tr_hillpower** = 2



zombie: custom variables

- choose **zombie**
  - go to **custom data**
    - ♦ add **zombie_speed_halfmax**
    - ♦ add **zombie_speed_hillpower**
    - ♦ add **zombie_damage_halfmax**
    - ♦ add **zombie_damage_hillpower**



Unzip Session07_checkpoint3.zip in ./PhysiCell to get this code.

Checklist

- [x] Plan
- [x] Populate and compile template project
- [x] Edit domain
- [x] Edit microenvironment
  - [x] food
  - [x] quorum
  - [x] zombie
- [ ] Create cell definitions
  - [x] farmer
  - [x] prey
  - [x] zombie
- [x] Add custom variables
- [ ] ~~Add custom parameters~~
- [ ] Declare custom functions
- [ ] Implement custom functions
  - [ ] prey_phenotype
  - [ ] zombie_phenotype
- [ ] Assign custom functions
- [ ] ~~Initial cell placement~~
- [ ] ~~Create and assign coloring~~
- [ ] Compile and test!

declare custom functions

```
open custom.h

declare:

void prey_phenotype( Cell* pCell, Phenotype& p, double dt );
void zombie_phenotype( Cell* pCell, Phenotype& p, double dt );
```

Checklist

- ☑ Plan
- ☑ Populate and compile template project
- ☑ Edit domain
- ☑ Edit microenvironment
  - ☑ food
  - ☑ quorum
  - ☑ zombie
- ❑ Create cell definitions
  - ☑ farmer
  - ☑ prey
  - ☑ zombie
- ☑ Add custom variables
- ❑ ~~Add custom parameters~~
- ☑ Declare custom functions
- ❑ Implement custom functions
  - ❑ prey_phenotype
  - ❑ zombie_phenotype
- ❑ Assign custom functions
- ❑ ~~Initial cell placement~~
- ❑ ~~Create and assign coloring~~
- ❑ Compile and test!

prey phenotype

```
void prey_phenotype( Cell* pCell, Phenotype& p, double dt )
{
   // set birth rate
   double param0 = get_base_behaviors( pCell, "cycle entry");
   double s = get_single_signal( pCell, "food" );
   double param = param0 * linear_response_function( s , 0.0, 1.0 );
   set_single_behavior( pCell, "cycle entry" , param );

   // set speed
   param0 = get_base_behaviors( pCell, "migration speed");
   // s = get_single_signal( pCell, "food" );
   param = param0 * linear_response_function( s , 0.0, 1.0 );
   set_single_behavior( pCell, "migration speed" , param );

   // set zombification
   param0 = get_base_behaviors( pCell, "transform to zombie");
   s = get_single_signal( pCell, "damage" );
   double hm = get_single_behavior( pCell, "custom:prey_tr_halfmax");
   double hp = get_single_behavior( pCell, "custom:prey_tr_hillpower");
   param = param0 * Hill_response_function( s, hm, hp );
   set_single_behavior( pCell, "transform to zombie" , param );

   return;
}
```

Checklist

- [x] Plan
- [x] Populate and compile template project
- [x] Edit domain
- [x] Edit microenvironment
  - [x] food
  - [x] quorum
  - [x] zombie
- [ ] Create cell definitions
  - [x] farmer
  - [x] prey
  - [x] zombie
- [x] Add custom variables
- [ ] ~~Add custom parameters~~
- [x] Declare custom functions
- [ ] Implement custom functions
  - [x] prey_phenotype
  - [ ] zombie_phenotype
- [ ] Assign custom functions
- [ ] ~~Initial cell placement~~
- [ ] ~~Create and assign coloring~~
- [ ] Compile and test!

zombie phenotype

```
void zombie_phenotype( Cell* pCell, Phenotype& p, double dt )
{
   // set speed
   // // need to add nubmer of nuclei to standard signals
   double param0 = get_single_base_behavior( pCell, "migration speed");
   double s = (double) pCell->state.number_of_nuclei;
   double hm = get_single_behavior( pCell, "custom:zombie_speed_halfmax");
   double hp = get_single_behavior( pCell, "custom:zombie_speed_hillpower");
   double param = param0 * (1-Hill_response_function(s,hm,hp) );
   set_single_behavior( pCell, "migration speed" , param );

   // set attack
   // // need to add damage rate to standard behaviors
   static Cell_Definition* pCD = find_cell_definition( pCell->type_name );
   param0 = pCD->phenotype.cell_interactions.damage_rate;
   s = (double) pCell->state.number_of_nuclei;
   param = param0 * s;
   p.cell_interactions.damage_rate = param;

   // set death
   s = get_single_signal( pCell, "damage");
   hm = get_single_behavior( pCell, "custom:zombie_damage_halfmax");
   hp = get_single_behavior( pCell, "custom:zombie_damage_hillpower");
   param = 0.01 * Hill_response_function( s,hm,hp );
   set_single_behavior( pCell, "apoptosis" , param );

   return;
}
```

Checklist

- ☑ Plan
- ☑ Populate and compile template project
- ☑ Edit domain
- ☑ Edit microenvironment
  - ☑ food
  - ☑ quorum
  - ☑ zombie
- ❑ Create cell definitions
  - ☑ farmer
  - ☑ prey
  - ☑ zombie
- ☑ Add custom variables
- ❑ ~~Add custom parameters~~
- ☑ Declare custom functions
- ☑ Implement custom functions
  - ☑ prey_phenotype
  - ☑ zombie_phenotype
- ❑ Assign custom functions
- ❑ ~~Initial cell placement~~
- ❑ ~~Create and assign coloring~~
- ❑ Compile and test!

assign custom functions

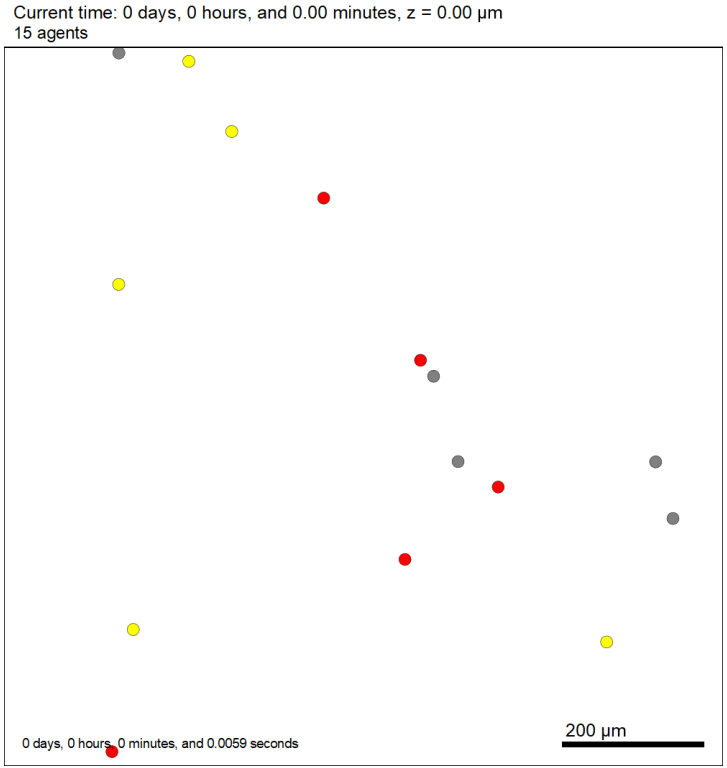
```
go to create_cell_types

   // ...
   cell_defaults.functions.contact_function = contact_function;

   Cell_Definition* pCD = find_cell_definition( "prey");
   pCD->functions.update_phenotype = prey_phenotype;

   pCD = find_cell_definition( "zombie");
   pCD->functions.update_phenotype = zombie_phenotype;

   /*
      This builds the map of cell definitions and summarizes the setup.
   */

   display_cell_definitions( std::cout );

   // ...
```

Checklist

- ☑ Plan
- ☑ Populate and compile template project
- ☑ Edit domain
- ☑ Edit microenvironment
  - ☑ food
  - ☑ quorum
  - ☑ zombie
- ❑ Create cell definitions
  - ☑ farmer
  - ☑ prey
  - ☑ zombie
- ☑ Add custom variables
- ❑ ~~Add custom parameters~~
- ☑ Declare custom functions
- ☑ Implement custom functions
  - ☑ prey_phenotype
  - ☑ zombie_phenotype
- ☑ Assign custom functions
- ❑ ~~Initial cell placement~~
- ❑ ~~Create and assign coloring~~
- ❑ Compile and test!

compile and test

- make
- ./project



Unzip Session07_checkpoint4.zip in ./PhysiCell to get this code.

better test from GUI

- Set to 50 of each cell type
- set max time to 2880 min
- go to **run** tab
  - set **exec** name to **project**
- click run

Coming in later sessions

Today

- Session 8: 3D visualization in Simularium (optional, Allen Cell Institute)
- Session 9: 3D visualization in FURY (optional, async / overnight)
- Session 10: Intro to PhysiBoSS (async / overnight)

Wednesday

- Session 11: PhysiBoSS (Institut Curie)
- Session 12: Intro to libRoadrunner
- Session 13: libRoadrunner example
- Session 14: model sharing on NanoHUB (optional, async / overnight
- Session 15: pressure and contact functions (async / overnight)

Funding Acknowledgements

PhysiCell Development:

- Breast Cancer Research Foundation
- Jayne Koskinas Ted Giovanis Foundation for Health and Policy
- National Cancer Institute (U01CA232137)
- National Science Foundation (1720625, 1818187)

Training Materials:

- Administrative supplement to NCI U01CA232137 (Year 2)

Other Funding:

- NCI / DOE / Frederick National Lab for Cancer Research (21X126F)
- DOD / Defense Threat Reduction Agency (HDTRA12110015)
- NIH Common Fund (3OT2OD026671-01S4)