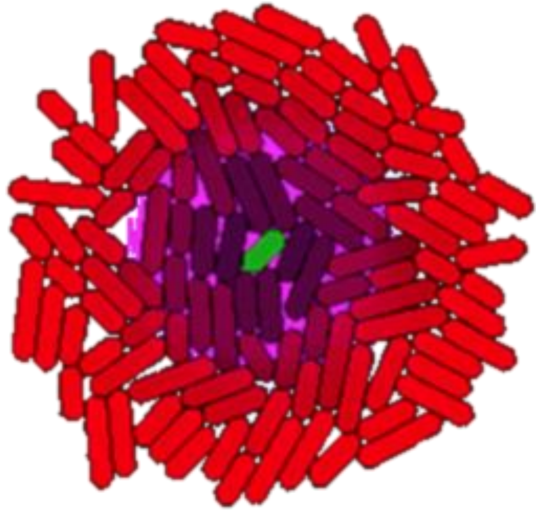


gro

Tutorial 3

Signals

- Declaring Signals
- Sending Signals
- Sensing Signals
- Absorbing Signals
- Reaction-Diffusion
- Bioprocessing example

Declaring Signals

This line defines a signal called “ahl”.

signal() takes two arguments:
degradation rate
diffusion rate

```
include gro

ahl := signal(5, 0.1);
```

By itself, declaring a signal doesn’t do anything! It only defines a signal for use later in the program, either by a cell or the environment (main loop). Don’t set the diffusion rate too high or you will run into numerical integration errors.

Sending Signals with Cells

```
include gro

ahl := signal(5, 0.1);

program signaler() := {
  true : {
    emit_signal(ahl,0.2)
  }
};

ecoli ( [], program signaler() );
```

A cell can send a signal using the emit_signal() function. It has two arguments:

The signal to use

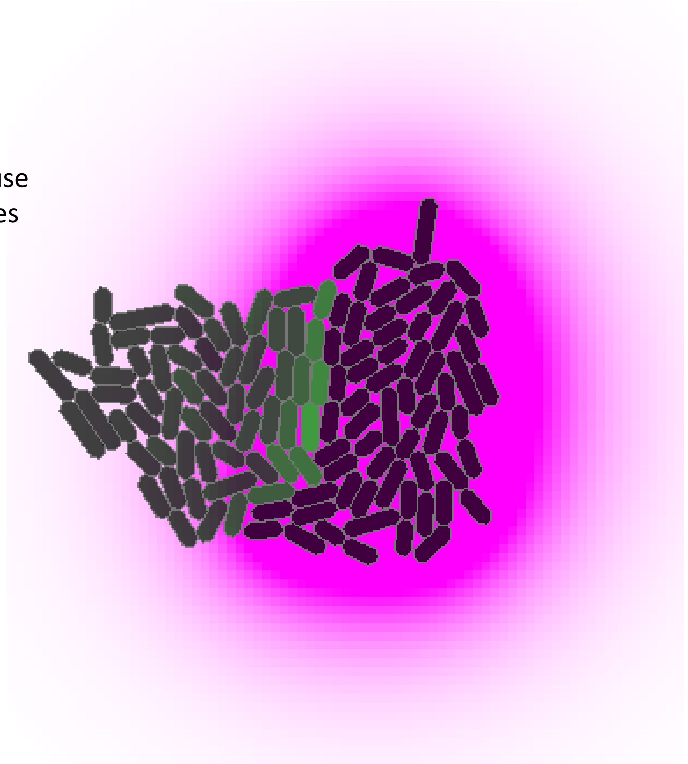
How much to release

These cells constantly emit signal. Try varying the parameters – what happens as you vary the diffusion and degradation rates of the signal, but keep their ratio the same?

Receiving Signals

```
include gro

ahl := signal(5, 0.1);
k := 2; // reporter scaling factor

program signaler() := {
  true : {
    emit_signal(ahl,0.2)
  }
};

program receiver() := {
  gfp := 0;
  rate(k*get_signal(ahl)) : {
    gfp := gfp + 1
  }
};

ecoli ( [x:=50,theta:=3.14/2], program signaler() );
ecoli ( [x:=-50], program receiver() );
```

To have a cell sense a signal, use get_signal(). This function takes one argument: the signal to detect.



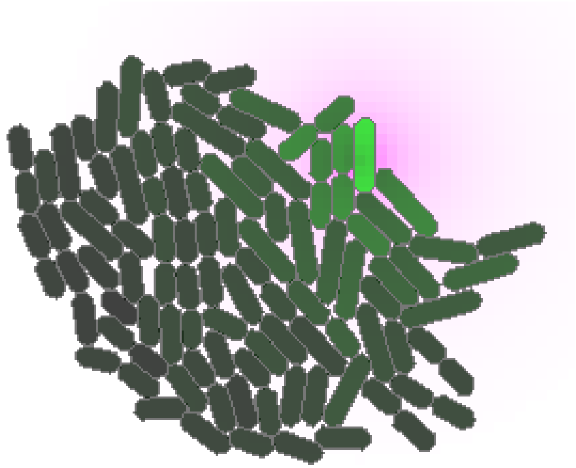
This program has two cell types. The new cell type, “receiver”, produces gfp at a rate proportional to the signal it receives

Setting Environment Signals

```
include gro

ahl := signal(5, 0.1);
k := 10; // reporter scaling factor

program receiver() := {
  gfp := 0;
  rate(k*get_signal(ahl)) : {
    gfp := gfp + 1
  }
};

program main() := {
  true : {
    set_signal(ahl,50,-50,1)
  }
};

ecoli ( [], program receiver() );
```

When defining a signal in main, use set_signal(). set_signal() takes four arguments:

- the signal to set
- x coordinate
- y coordinate
- the amount of signal to release



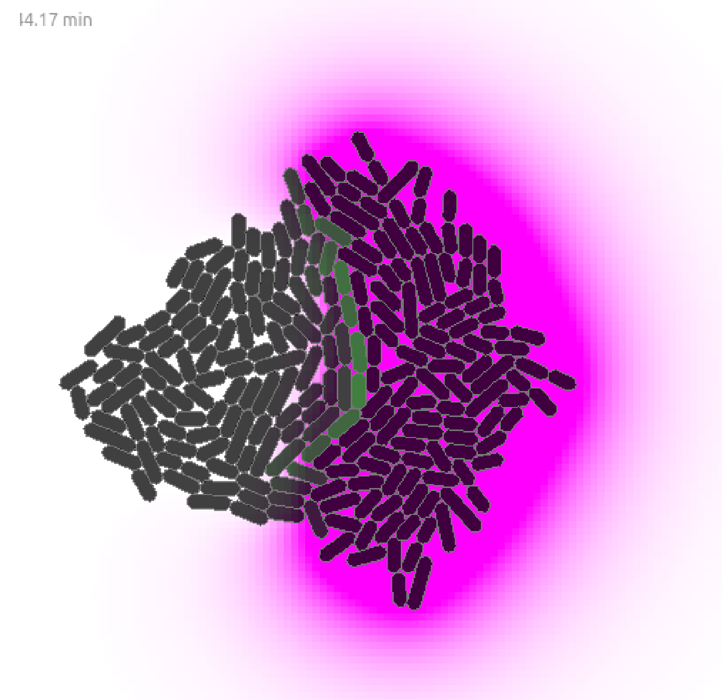
Coordinates in gro have the origin in the center of the screen, with x coordinates increasing from left to right and y coordinates increasing from top to bottom.

Absorbing Signals

```
include gro

ahl := signal(5, 0.1);
k := 2; // reporter scaling factor

program signaler() := {
  true : {
    emit_signal(ahl,0.2)
  }
};

program receiver() := {
  gfp := 0;
  true : {
    absorb_signal(ahl,0.1)
  }
  rate(k*get_signal(ahl)) : {
    gfp := gfp + 1
  }
};

ecoli ( [x:=50,theta:=3.14/2], program signaler() );
ecoli ( [x:=-50], program receiver() );
```

To have cells absorb a signal, use absorb_signal(), which takes two arguments:

- signal to absorb
- how much signal to absorb



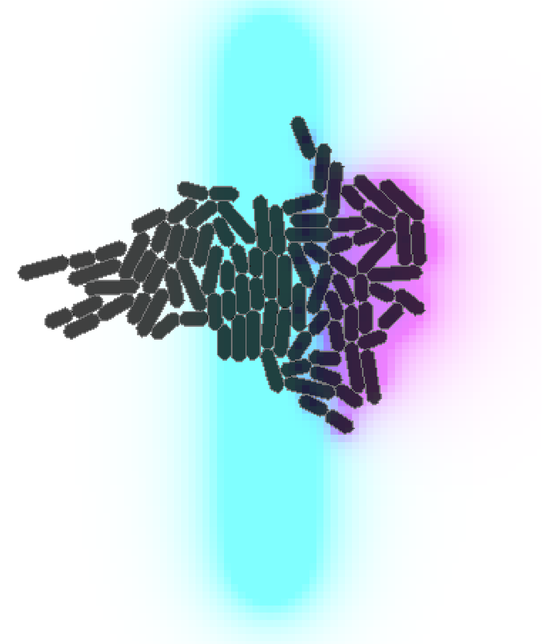
This program is identical to the receiving signals program, but receiver cells eat up the signal. Absorption is useful both for accuracy and multicellular behaviors: cells that eat up a nutrient signal should absorb it, and signal removal is found in many natural multicellular signaling circuits.

Reaction-Diffusion

```
include gro

ahl := signal(5, 0.1);
antiahl := signal(1, 0.1);
reaction({ahl,antiahl},{antiahl},10);

program signaler() := {
  true : {
    emit_signal(ahl,2)
  }
};

program receiver() := {
  gfp := 0;
  rate(get_signal(ahl)) : {
    gfp := gfp + 1
  }
};

program main() := {
  true: {
    foreach i in range 10 do {
      set_signal(antiahl,0,(200-40*i),10)
    } end;
  }
};
ecoli ( [x:=50,theta:=3.14/2], program signaler() );
ecoli ( [x:=-50], program receiver() );
```

reaction() defines how signals interact and takes 3 arguments:

- A list of reactants
- A list of products
- The reaction rate



Reaction-diffusion reactions are based on chemicals that can (1) react with each other (or themselves) and (2) diffuse. Basic pattern formation can be generated via reaction-diffusion alone.

It can also be used for simpler behaviors: this program is identical to the receiving signals program, but with a line of “anti-ahl” signal that destroys ahl separating sending and receiving cells. How does this change the behavior of the receiver cells?

Example: Bioprocessing

Biomass is red, enzyme is green, food is blue

In this simulation, food (and therefore growth) can only come from degradation of the biomass via an excreted enzyme

Cell growth rate depends on nutrient availability

Uneven distribution of non-diffusing feedstock

```
include gro

set_theme ( bright_theme << [ signals := { { 1,0,0 }, { 0,1,0 } , { 0,0,1 } } ] );
biomass := signal(0, 0);
enzyme := signal(4,0.3);
food := signal(5, 0.1);
reaction({biomass,enzyme},{food,enzyme},5);
set("ecoli_growth_rate",0.0);

program bioprocessor() := {
  true : {
    set("ecoli_growth_rate",get_signal(food)),
    emit_signal(enzyme,1)
  }
};

program main() := {
  t := 0;
  true: { t := t + dt }
  foreach i in range 2000 do {
        set_signal(biomass,rand(400),(rand(800)-400),10)
  } end;
};

ecoli ( [], program bioprocessor() );
```

Run the code! See what happens to the growing cell distribution by changing the diffusion rates.

Example: Bioprocessing