

Advanced Session 3: PhysiCell coloring functions

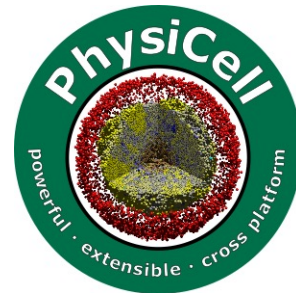
John Metzcar



@jmetzcar

PhysiCell Project

August 7, 2023 (asynchronous session)



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

 **@PhysiCell**

Goals and links

- Built in SVG visualization in PhysiCell
- Demonstrate default cell coloring function from template project
- Explore “on the fly” coloring in PhysiCell Studio
- Create custom coloring function
- Learning goal
 - Ability to create a custom cell coloring function
- Code and video links
 - [Code](#) – full code
 - [Code](#) – project only
 - [Video](#)

Default PhysiCell visualization: SVGs

SVG – Scalable Vector Graphics: XML based image format



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

 **@PhysiCell**

Default PhysiCell visualization: SVGs

Graphic: Cells – size, location, and color

Current time: 3 days, 9 hours, and 0.00 minutes, z = 0.00 μm
85 agents



XML-based

```
<!-- Created with PhysiCell
(http://PhysiCell.MathCancer.org/) -->
<svg
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:cc="http://creativecommons.org/ns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
  xmlns:svg="http://www.w3.org/2000/svg"
  xmlns="http://www.w3.org/2000/svg"
  version="1.1"
  width="1000"
  height="1070"
  id="svg2">
<rect x="0" y="0" width="1000" height="1070" stroke-
width="2" stroke="white" fill="white"/>
<text x="12.5" y="30"
  font-family="Arial" font-size="25" fill="black" >
Current time: 3 days, 9 hours, and 0.00 minutes, z =
0.00 &#956;m
</text>
<text x="12.5" y="57.5"
  font-family="Arial" font-size="23.75" fill="black" >
85 agents
</text>
<g id="tissue"
  transform="translate(0,1070) scale(1,-1)">
<g id="ECM">
</g>
<g id="cells">
<g id="cell0" type="default" dead="false" >
<circle cx="186.241" cy="965.219" r="9.98041"
  stroke-width="0.5" stroke="black"
  fill="rgb(255,255,0)"/>
<circle cx="186.241" cy="965.219" r="6.05464"
  stroke-width="0.5" stroke="rgb(255,255,0)"
  fill="rgb(255,255,0)"/>
</g>
```



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

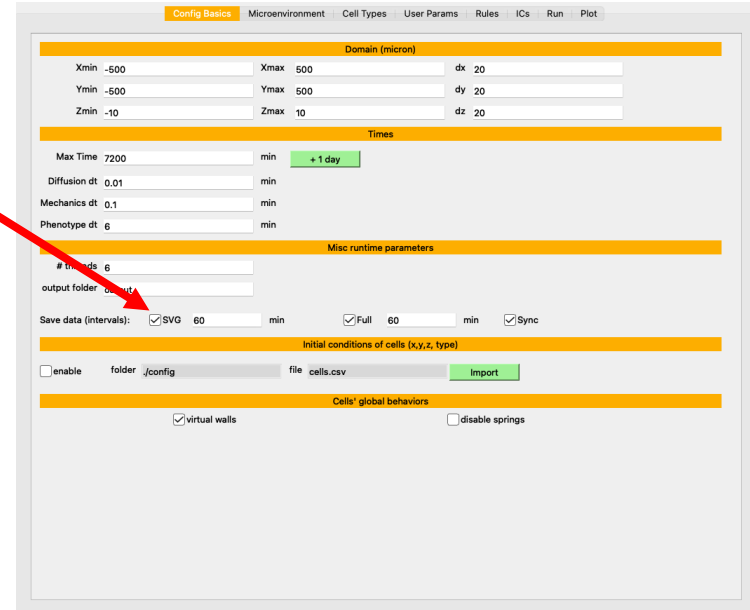
PhysiCell Project

PhysiCell.org

@PhysiCell

Default PhysiCell visualization

- SVGs output by default



The screenshot shows the 'Config Basics' tab of the PhysiCell configuration window. The 'Save data (intervals):' section is highlighted with a red arrow. The 'SVG' checkbox is checked, indicating that SVGs are output by default. Other settings include 'Full' and 'Sync' checkboxes, both of which are also checked. The 'Initial conditions of cells (x,y,z, type)' section shows 'enable' checked, 'folder' set to 'jconfig', 'file' set to 'cells.csv', and an 'Import' button. The 'Cells' global behaviors section shows 'virtual walls' checked and 'disable springs' unchecked.

Domain (micron)		
Xmin -500	Xmax 500	dx 20
Ymin -500	Ymax 500	dy 20
Zmin -10	Zmax 10	dz 20

Times	
Max Time 7200	min + 1 day
Diffusion dt 0.01	min
Mechanics dt 0.1	min
Phenotype dt 6	min

Misc runtime parameters	
# threads 6	
output folder output	
Save data (intervals):	<input checked="" type="checkbox"/> SVG 60 min <input checked="" type="checkbox"/> Full 60 min <input checked="" type="checkbox"/> Sync

Initial conditions of cells (x,y,z, type)	
<input type="checkbox"/> enable	folder jconfig file cells.csv Import

Cells' global behaviors	
<input checked="" type="checkbox"/> virtual walls	<input type="checkbox"/> disable springs



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

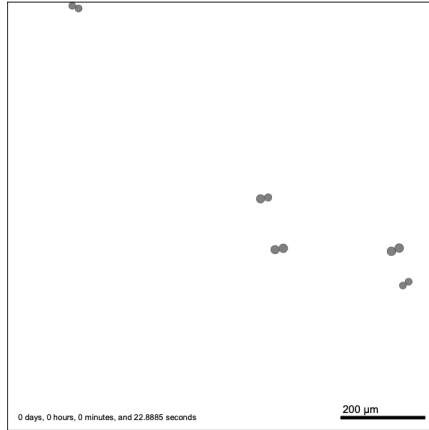
PhysiCell.org

@PhysiCell

Default PhysiCell visualization

- SVGs output by default
- Function generates visual output of cell positions and other simulation metadata

Current time: 1 days, 0 hours, and 0.00 minutes, z = 0.00 μm
10 agents

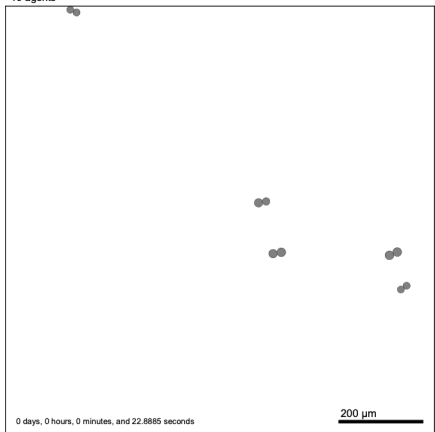


Output at 1440 minutes
of template project

Default PhysiCell visualization

- SVGs output by default
- Function generates visual output of cell positions and other simulation metadata
- Also produces a legend with cell names

Current time: 1 days, 0 hours, and 0.00 minutes, z = 0.00 μm
10 agents



Output at 1440 minutes
of template project

● default

Separate SVG cell
legend

Config Basics | Microenvironment | Cell Types | User Params | Rules | ICs | Run | Plot

Domain (micron)

Xmin -500	Xmax 500	dx 20
Ymin -500	Ymax 500	dy 20
Zmin -10	Zmax 10	dz 20

Times

Max Time 7200 min + 1 day

Diffusion dt 0.01 min

Mechanics dt 0.1 min

Phenotype dt 6 min

Misc runtime parameters

threads 6

output folder output

Save data (intervals): ☒ SVG 60 min ☒ Full 60 min ☒ Sync

Initial conditions of cells (x,y,z, type)

☐ enable folder jconfig file cells.csv Import

Cells' global behaviors

☒ virtual walls ☐ disable springs

Modifying cell coloring: Pre-built and custom



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

 **@PhysiCell**

Modifying cell coloring: Pre-built and custom

- Can use cell coloring to identify cell properties
 - Type
 - Cell cycle phase
 - Pressure
 - Basically – ANY variable (continuous, categorical, anything) associated with an agent

Modifying cell coloring: Pre-built and custom

- Can use cell coloring to identify cell properties
 - Type
 - Cell cycle phase
 - Pressure
 - Basically – ANY variable (continuous, categorical, anything) associated with an agent
- Two+ flavors:
 - Pre-built
 - Custom
 - Pre-built + custom
 - The Studio

Pre-built coloring functions

- PhysiCell_pathology.cpp

- Contains a number of pre-built coloring functions

```
std::vector<std::string> simple_cell_coloring( Cell* pCell ); // done  
std::vector<std::string> false_cell_coloring_Ki67( Cell* pCell ); // done  
std::vector<std::string> false_cell_coloring_live_dead( Cell* pCell ); // done  
std::vector<std::string> false_cell_coloring_cycling_quiescent( Cell* pCell ); // done  
std::vector<std::string> false_cell_coloring_cytometry( Cell* pCell );  
std::vector<std::string> hematoxylin_and_eosin_cell_coloring( Cell* pCell ); // done  
std::vector<std::string> hematoxylin_and_eosin_stroma_coloring( double& ECM_fraction , double& blood_vessel_fraction); // planned  
std::vector<std::string> paint_by_number_cell_coloring( Cell* pCell ); // done
```

- Default Template coloring function: **paint_by_number_cell_coloring**

Highlights of paint_by_number function (1)

```
std::vector<std::string> paint_by_number_cell_coloring( Cell* pCell )
{
    static std::vector< std::string > colors(0);
    static bool setup_done = false;
    if( setup_done == false )
    {
        colors.push_back( "grey" ); // default color will be grey

        colors.push_back( "red" );
        colors.push_back( "yellow" );
        colors.push_back( "green" );
        colors.push_back( "blue" );
        colors.push_back( "magenta" );
        colors.push_back( "orange" );
        colors.push_back( "lime" );
        colors.push_back( "cyan" );
        colors.push_back( "hotpink" );
        colors.push_back( "peachpuff" );
        colors.push_back( "darkseagreen" );
        colors.push_back( "lightskyblue" );

        setup_done = true;
    }
}
```



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

 @PhysiCell

Highlights of paint_by_number function (2)

```
std::vector<std::string> output = { "black", "black", "black", "black" };  
// paint by number -- by cell type  
std::string interior_color = "white";  
if( pCell->type < 13 )  
{ interior_color = colors[ pCell->type ]; }  
output[0] = interior_color; // set cytoplasm color
```

Eventually all values in output get assigned to interior OR by death state (it colors by apoptosis and necrosis)

```
return output;  
}
```

- There are four colors – for nucleus, cytoplasm, and each border
 - We cover only uniform coloring in this session but ...
 - {cytoplasm, cytoplasm border, nucleus, nucleus border}
 - See PhysiCell_pathology.cpp and PhysiCell_SVG.cpp for more information

Demo results of paint by number

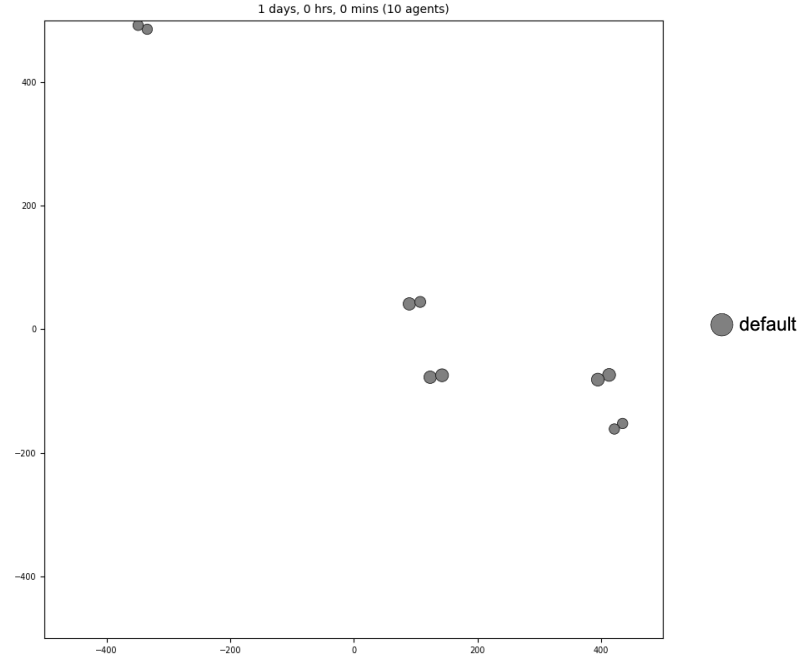
Run commands below in PhysiCell
folder to start the Studio with the
template project

```
make reset // if needed
```

```
make template
```

```
// Start Studio
```

```
your_path_to_studio/bin/studio.py -e project -c  
config/PhysiCell_settings.xml
```



Default output – grey cells
1 simulated day



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

@PhysiCell

Demo results of paint by number

Within the Studio-

Go to Cell Types tab

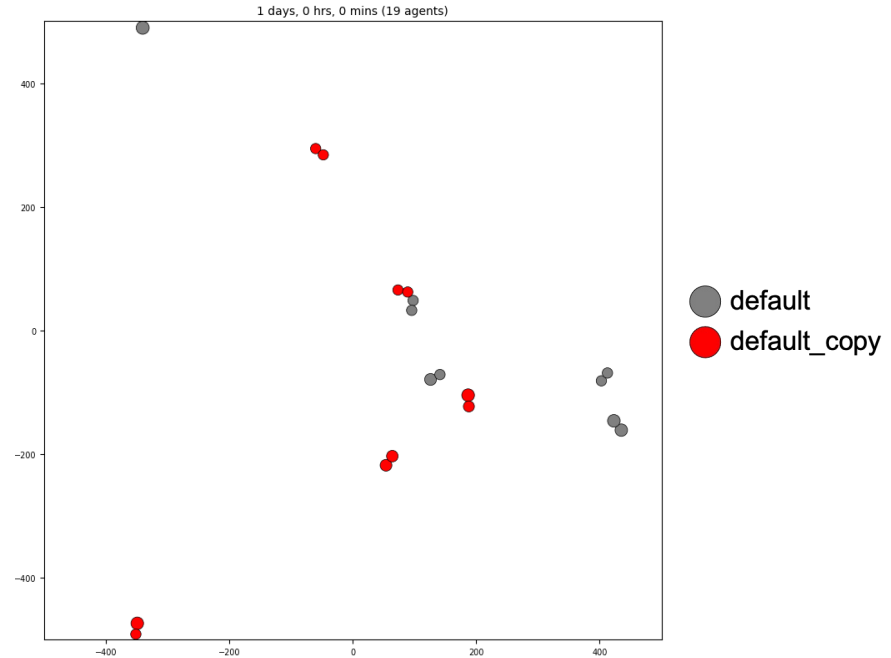
Click “Copy”

Place cursor in cell type field

Change name to “default_copy”

Run simulation

- Now see red cells
- Coloring based on categorical variable “type”
- Expected based on our coloring function
“red” is after “grey”



Modified output – grey and red cells
1 simulated day



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

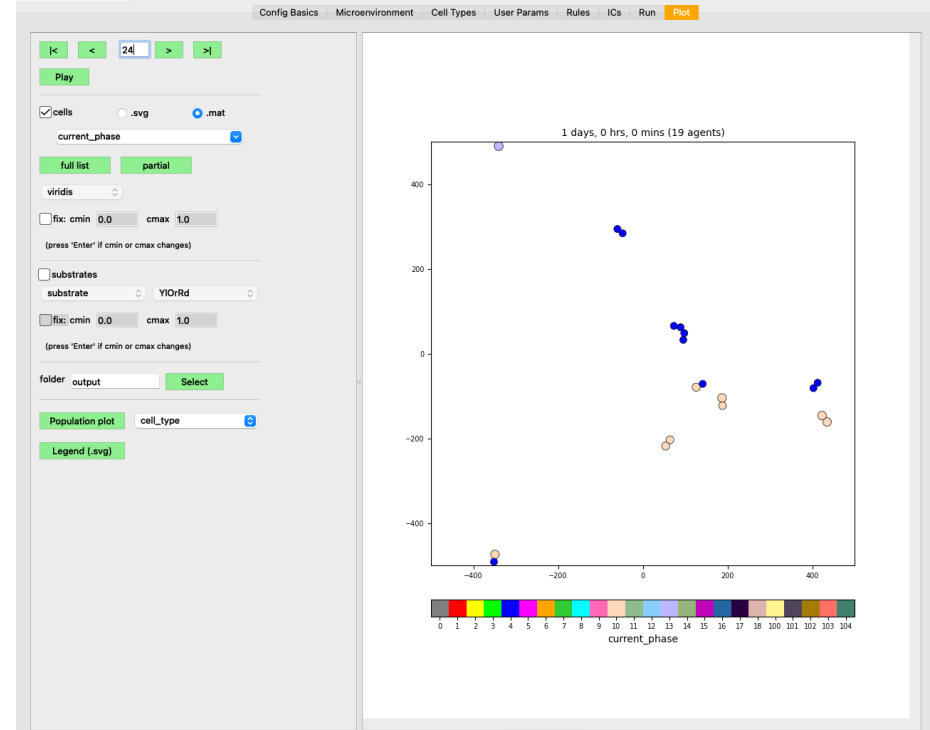
PhysiCell Project

PhysiCell.org

@PhysiCell

Coloring in the Studio

- Can use the studio for a quick visualization
 - Motility direction
 - Pressure
 - Many more properties
- Useful for (among other things), exploring relevant parameters to visualize
- Click the .mat button
 - Select by typing or drop down
 - click full list to see all properties



Coloring by cycle phase - categorical

Custom coloring function (1)



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

 **@PhysiCell**

Custom coloring function (1)

In custom.h file

```
std::vector<std::string> custom_coloring_function( Cell* pCell);
```

In custom.cpp file

```
std::vector<std::string> custom_coloring_function( Cell* pCell)
{
    // color 0: cytoplasm fill
    // color 1: outer outline
    // color 2: nuclear fill
    // color 3: nuclear outline
    // start with color-by-number
    // dead cells: black if apoptotic, brown if necrotic – will come from paint by number
    // live cells: shade by proliferation rate
}
```

Custom coloring function (2)

```
std::vector<std::string> custom_coloring_function( Cell* pCell)
{
    // start with color-by-number

    std::vector<std::string> output = paint_by_number_cell_coloring(pCell);

    // dead cells: black if apoptotic, brown if necrotic – coming from paint by number
    // live cells: shade by proliferation rate
}
```



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

 @PhysiCell

Custom coloring function (3)

```
std::vector<std::string> custom_coloring_function( Cell* pCell)
{
    std::vector<std::string> output = paint_by_number_cell_coloring(pCell);
    if( pCell->type_name== "default" && pCell->phenotype.death.dead == false ) // live tumor cells: shade by proliferation rate
    {
        // get relative birth rate
        double s = 10 * get_single_behavior( pCell, "cycle entry" )
            / get_single_base_behavior( pCell, "cycle entry" );
        if( s > 1 )
        { s = 1; }
        // make color
        int color = (int) round( 255.0 * s );
        char szColor[1024];
        // interpolate from blue to yellow
        sprintf( szColor, "rgb(%u,%u,%u)",color,color,255-color );
        // modify output

        output[0] = szColor;
        output[2] = szColor;
        output[3] = szColor;
    }
    return output;
}
```

Custom coloring function (4)

In main.cpp

```
std::vector<std::string> (*cell_coloring_function)(Cell*) =  
    custom_coloring_function;
```



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

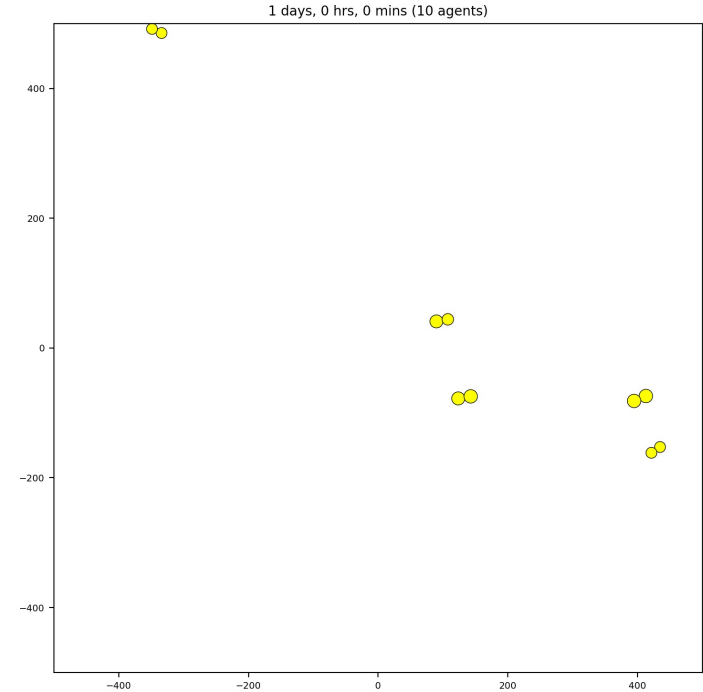
PhysiCell Project

PhysiCell.org

 **@PhysiCell**

Demo of new coloring function

- Run **make**
- Rerun simulation (in Studio or CLI)
- RGB = (255, 255, 0) = Yellow



All cells have cell cycle at base level



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

@PhysiCell

Final notes

- Pre-built coloring functions – in PhysiCell_pathology.cpp



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

 **@PhysiCell**

Final notes

- Pre-built coloring functions – in PhysiCell_pathology.cpp
- Default coloring function in template – “paint by type”
 - Colors by cell type



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project

PhysiCell.org

 **@PhysiCell**

Final notes

- Pre-built coloring functions – in PhysiCell_pathology.cpp
- Default coloring function in template – “paint by type”
 - Colors by cell type
- Exploring attributes to color with – The Studio

Final notes

- Pre-built coloring functions – in PhysiCell_pathology.cpp
- Default coloring function in template – “paint by type”
 - Colors by cell type
- Exploring attributes to color by – The Studio
- Custom coloring functions
 - Can color by any cell attribute (similar to the Studio)
 - Can color both cytoplasm and nucleus
 - ◆ Example – could coloring by both type AND cell cycle entry rate – left to excited viewer!

Final notes

- Pre-built coloring functions – in PhysiCell_pathology.cpp
- Default coloring function in template – “paint by type”
 - Colors by cell type
- Exploring attributes to color by – The Studio
- Custom coloring functions
 - Can color by any cell attribute (similar to the Studio)
 - Can color both cytoplasm and nucleus
 - ♦ Example – could coloring by both type AND cell cycle entry rate – left to excited viewer!
- Colors follow the W3C standards for SVG files
 - Names, RGB values, etc.
 - <https://www.w3.org/TR/SVG11/types.html#ColorKeywords>

Funding Acknowledgements



NATIONAL
CANCER
INSTITUTE



leidos



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)