

OpenVT – A Standardized Ecosystem for Virtual Tissue Simulation

Randy Heiland, Indiana University
IMAG Multiscale Modeling and Viral Pandemics WG
May 2, 2024

Core team: Bart Jardine, Pete Fyffe, Kyle Stirling, Michael Getz,
Hayden Fennell, Randy Heiland, PIs

Larger team: developers of any multicellular modeling framework

PIs: James Glazier, David Wild, Paul Macklin

<https://github.com/OpenVT/OpenVT-Edu> - see /slides

Outline

- NSF I-Corps
- Fundamental multicellular modeling approaches
- Existing frameworks (and features)
- Cellular Potts vs. Center-based
- Shareable “standards”
- Early effort: convert PhysiCell model to CC3D
- Reference models
- Reproducibility
- Discussion

NSF Pathways to Enable Open-Source Ecosystems (POSE II)

<https://new.nsf.gov/funding/opportunities/pathways-enable-open-source-ecosystems-pose>

- ~19 awards (in ~Sept 2023)
- *“harness the power of open-source development for the creation of new technology solutions to problems of national and societal importance”*
- Open source research → **sustainable** open source ecosystems
- In Dec 2023, NSF notified awardees that they would need to participate in the I-Corps program, starting in January 2024.

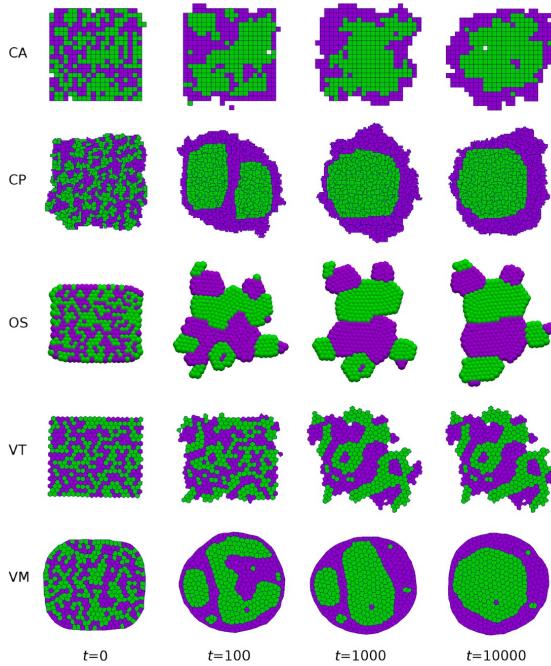
POSE, NSF I-Corps

<https://new.nsf.gov/funding/initiatives/i-corps>

- Entrepreneurial training program (virtual):
 - What is your product? Hypotheses for (broader) use?
 - Who are your stakeholders? (users, developers, funders, ...)
 - Governance. Sustainability. Milestones.
 - Elevator pitch. Business Model Canvas.
 - Learn their Airtable system for reporting, assignments.
- Normally: 7 weeks; 100 interviews. However, after feedback from awardees...
→ 4 weeks; 60 interviews

Personal takeaway: a LOT of effort, but do it if you get the chance.
The interviews alone are worth the effort.

Different fundamental modeling approaches



- CA – Cellular Automata
- CP – Cellular Potts \leftarrow CompuCell3D
- OS/CB – Overlapping Spheres, or Center-based
PhysiCell \uparrow
- VT – Voronoi tessellation
- VM – Vertex Method

Figure taken from:

Comparing individual-based approaches to modelling the self-organization of multicellular tissues (2017). Osborne, et. al.
doi:10.1371/journal.pcbi.1005387

Frameworks' capabilities and features

- OpenVT will maintain a *living* spreadsheet of frameworks and their features (<https://github.com/OpenVT/frameworks>)

	Class					PDE			Lng	APIs	GUI	Web	Intracellular					
	CA	CP	CB	VT	VM	Diffusion		Advection										
						isotropic	anisotropic											
CompuCell3D	✓					✓			C++	Python	✓	*1	✓					
PhysiCell		✓				✓			C++	C++	✓	*2	✓					
Morpheus																		
Chaste																		
Tissue Forge																		
Artistoo																		
HAL																		
Biocellion																		

Published tables of frameworks+features (1)

Feature
On-Lattice ABMs
Off-Lattice Point ABMs
Off-Lattice Spherical ABMs
Voronoi Tesselation ABMs
Cellular Potts ABMs
Multinomial Population ABMs [35]
Diffusion PDEs
Advection PDEs
SBML Compatible [36]
Real-Time Visualization
Single-Model Parallelization
Windows Compatible
Mac Compatible
Linux Compatible
User-Facing Programming Language*

Table 1

Comparison of HAL with other agent-based modeling frameworks commonly used in tissue modeling as of December 2019.

Features are marked according to whether there exists a built-in user accessible implementation.

Feature	HAL	Cha	Rep	Mas	Net	Phy	Cel	Bio	Tim	Yal	Epi	Com	Mor	TST
On-Lattice ABMs	✓	✓	✓	✓	✓									✓
Off-Lattice Point ABMs	✓	✓	✓	✓	✓									
Off-Lattice Spherical ABMs	✓	✓				✓	✓	✓	✓	✓	✓			
Voronoi Tesselation ABMs	✓	✓												
Cellular Potts ABMs	✓											✓	✓	✓
Multinomial Population ABMs [35]	✓													
Diffusion PDEs	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Advection PDEs	✓	✓					✓					✓	✓	
SBML Compatible [36]						✓					✓	✓	✓	
Real-Time Visualization	✓		✓	✓	✓		✓				✓		✓	✓
Single-Model Parallelization		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Compatible	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓
Mac Compatible	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
Linux Compatible	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
User-Facing Programming Language*	J	C	R	J	N	C	C	C	C	U	I	P	M	C

Framework Abbreviations: **HAL**:Hybrid Automata Library, **Cha**:Chaste, **Rep**:Repast, **Mas**:Mason, **Net**:Netlogo, **Phy**:Physicell, **Cel**:CellSys, **Bio**:Biocellion, **Tim**:Timothy, **Yal**:Yalla, **Epi**:Episim, **Com**:CompuCell3D, **Mor**:Morpheus, **TST**:Tissue Simulation Toolkit

* User-Facing Programming Languages: J: Java, C: C/C++, R: ReLogo/Java/Groovy, N: NetLogo Programming Language, U: CUDA/C++, I: Graphical Interface, P: Python/XML, M: Morpheus model description language

I suggest using a
hierarchical/categorical spreadsheet

- Hybrid Automata Library: A flexible platform for hybrid modeling with real-time visualization (2020). Bravo, et. al. <https://doi.org/10.1371/journal.pcbi.1007635>

Published tables of frameworks+features (2)

TABLE 1. Computational Methods and Open Source Toolkits

Method	Open Source Toolkits
Cellular automata	Chaste: http://www.cs.ox.ac.uk/chaste NetLogo: https://ccl.northwestern.edu/netlogo Repast: https://repast.github.io
Lattice gas cellular automata	NetLogo: https://ccl.northwestern.edu/netlogo
Cellular Potts	CompuCell3D: http://www.compucell3d.org EPISIM: http://tigacenter.bioquant.uni-heidelberg.de/episim.html Morpheus: https://imc.zih.tu-dresden.de/wiki/morpheus/doku.php Tissue Simulation Toolkit: https://biomodel.project.cwi.nl/software/software#TST
Vertex based	Chaste: http://www.cs.ox.ac.uk/chaste Tyssue: https://github.com/DamCB/tyssue
Center based	Biocellion: * https://biocellion.com Chaste: http://www.cs.ox.ac.uk/chaste FLAME: http://flame.ac.uk PhysiCell: http://PhysiCell.MathCancer.org Timothy: https://timothy.icm.edu.pl
Immersed boundary	IBCell: * https://labpages.moffitt.org/rejniakk/LabsTools.html#LabIAD
Other	BioFVM: http://BioFVM.MathCancer.org Microvessel Chaste: https://jmsgrogan.github.io/MicrovesselChaste VCell: http://vcell.org

- A Review of Cell-Based Computational Modeling in Cancer Biology (2018).
Metzcar, et. al. <https://doi.org/10.1200/CCI.18.00069>

Published tables of frameworks+features (3)

Table 7. Appropriate models. Level of appropriateness of model: **highly appropriate**—model was developed in order to investigate this mechanism; **appropriate**—model can be used for this mechanism with minimal effort; **less appropriate**—model can only be used with this mechanism through careful tuning of parameters to match more appropriate models. Where appropriate advantages and disadvantages of each model in each example are given.

Example	CA	CP	OS	VT	VM
Adhesion	Appropriate: required Potts-like extension for differential adhesion	Highly appropriate: this is the model problem for the CP model	Less appropriate: unnatural differential adhesion, no engulfment	Less appropriate: unnatural differential adhesion, no engulfment	Highly appropriate: a model problem for the VM
Proliferation	Less appropriate: no cell compression	Appropriate: motility parameters tuned for stability	Highly appropriate: similarities to the VT model give same advantages	Highly appropriate: established paradigm for crypt models, previously fit to data	Appropriate: boundary conditions influence natural drift
Short-range signalling	Less appropriate: lattice artifacts, instantaneous neighbour transition	Less appropriate: lattice artifacts, excessive stochasticity with chosen parameters (required for mechanical stability)	Highly appropriate: smooth neighbour transition, copes with high levels of compression	Highly appropriate: smooth neighbour transition, copes with high levels of compression	Appropriate: parameter choice to prevent inverted elements from hyper proliferation
Long-range signalling	Less appropriate: costs for calculating FE mesh, lattice artifacts, extra transport of morphogen with movement	Appropriate: costs for calculating FE mesh	Appropriate: costs for calculating FE mesh	Highly appropriate: ready built FE mesh with 1-1 correspondence between cells and nodes	Highly appropriate: FE mesh generation from cells, FE mesh more refined than cells

- Comparing individual-based approaches to modelling the self-organization of multicellular tissues (2017). Osborne, et. al. <https://doi.org/10.1371/journal.pcbi.1005387>

Initial focus on Cellular Potts and Center-based

How are cells defined? Dynamics?

Cellular Potts (CP): CC3D

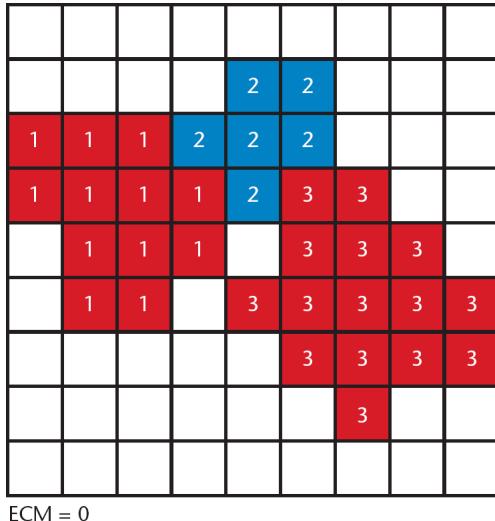
- Pixels on a lattice
- Inherent morphology
- Minimize total “energy”; Hamiltonian (global calculation)
- No explicit time, use MCS
- No explicit spatial units, use pixels
- Medium: always a cell type

Center-based (CB): PhysiCell

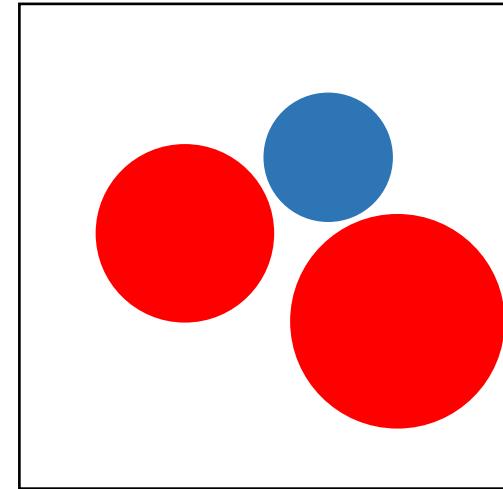
- Center, volume (lattice-free)
- Classic physics-based dynamics (local calculation), e.g., potential functions for adhesion and repulsion; spring mechanics
- Explicit time (e.g., mins)
- Explicit spatial units (e.g., μm)
- No Medium cell type

Cells: spatial definition

- CP (CC3D): pixels on a lattice
- CB (PhysiCell): center + volume (lattice-free)



Multi-Scale Modeling of Tissues Using CompuCell3D(2018), Swat et al., <https://doi.org/10.1016/B978-0-12-388403-9.00013-8>



PhysiCell: an Open Source Physics-Based Cell Simulator for Multicellular Systems (2018), Ghaffarizadeh et. al., PLoS Comput. Biol. <https://doi.org/10.1371/journal.pcbi.1005991>

OpenVT: What might we “standardize”?

- Data formats for initial conditions (of cells and substrates)
- Concept of a cell’s phenotype?
 - May contribute to “standard” model spec
 - May contribute to digital cell lines
 - → MultiCellDS (rf. later slide)

Standard format for Initial Conditions (ICs)?

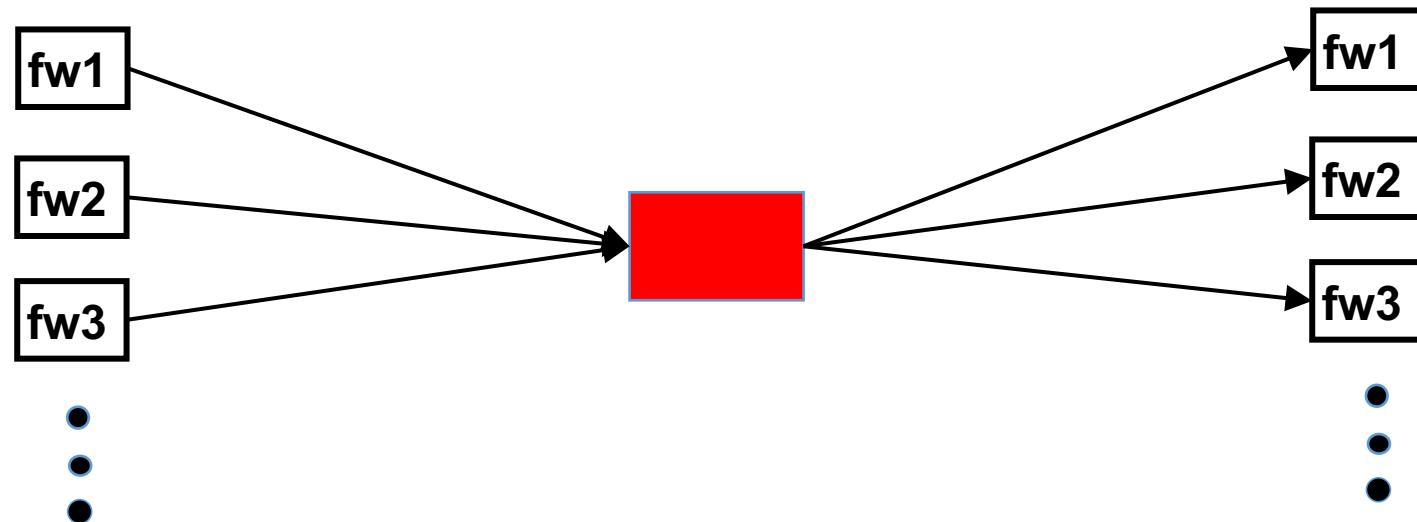
- Don't require adoption in each framework, just agreement for being able to share/convert between frameworks.
- What type of data do we have?
 - Cells: cell type, {centroid vs. lattice point(s)}, [scalar fields], [vector fields?], [tensors?]
 - Substrates: mesh, concentrations

Standard, intermediate data format for cells' ICs

Each framework provides scripts to convert:

native format → standard format

standard format → native format.

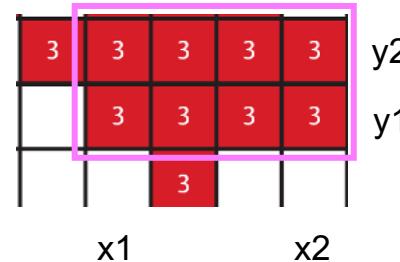


Cells' initial conditions native formats

- CC3D: PIFF (Potts Initial File)

https://compuccell3dreferencemanual.readthedocs.io/en/latest/pif_initializer.html

cell# celltype x1 x2 y1 y2 z1 z2



- PhysiCell: CSV

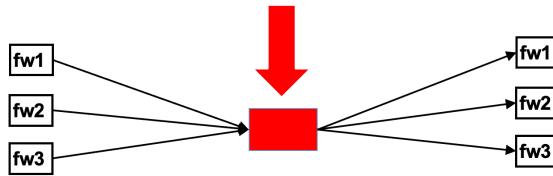
<https://github.com/MathCancer/PhysiCell/blob/master/changes.md#1110>

x,y,z,celltype[,...]

What type of data is needed for ICs?

- Cells:
 - Type (String vs. Int)
 - Centroid (x,y,z) vs. Lattice indices (i,j,k)
 - Scalars: volume, pressure, ...
 - Vector: velocity,
 - ...
- Substrates:
 - Regular(?) Mesh: dims, concentrations, gradient

Standard, intermediate file format for ICs?



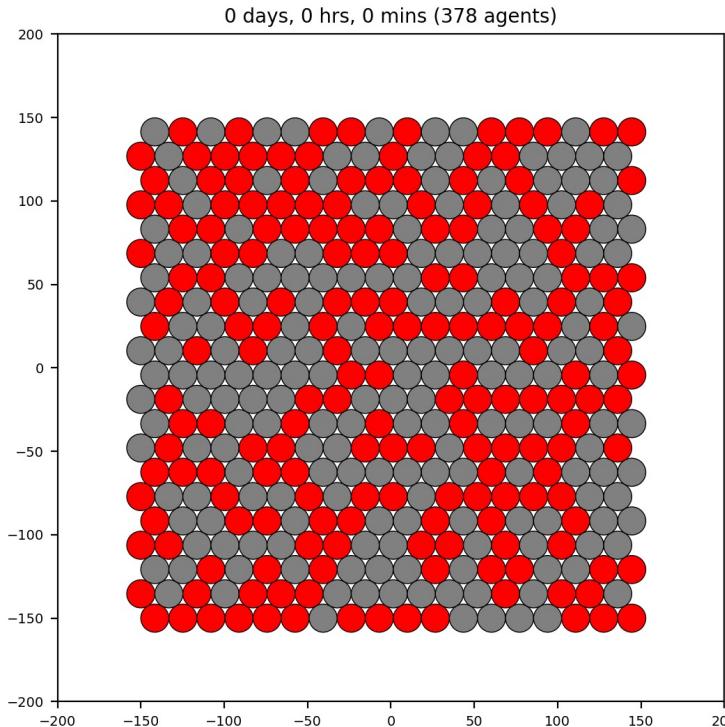
VTK Unstructured Grid

- <https://vtk.org/doc/nightly/html/classvtkUnstructuredGrid.html>
- https://docs.vtk.org/en/latest/design_documents/VTKFileFormats.html
- <https://examples.vtk.org/site/VTKBook/05Chapter5/>

Blender - blender.org, wiki.blender.org/wiki/Main_Page

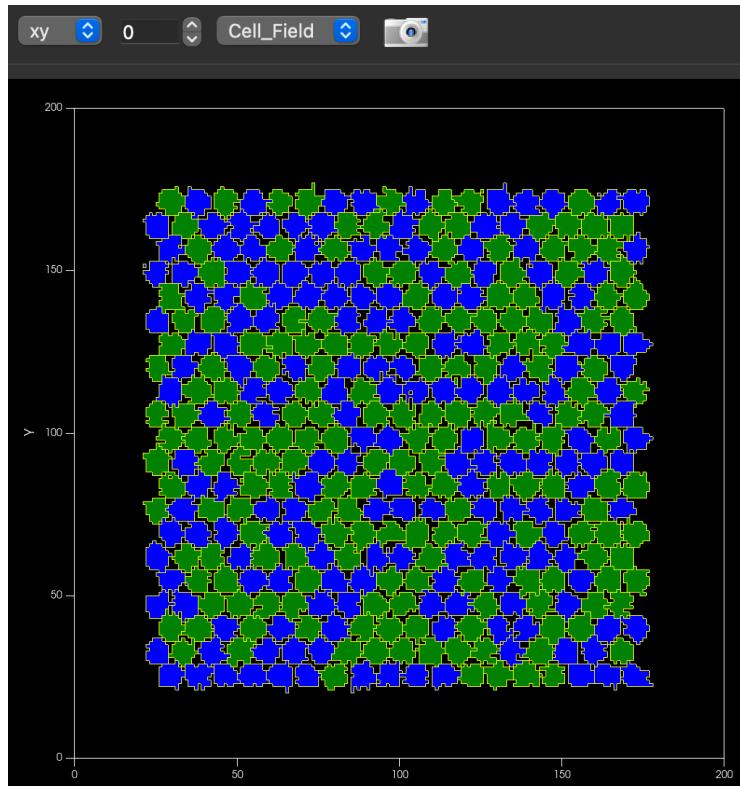
- <https://unstructured-grids.readthedocs.io/en/latest/ug.html>

Cells' ICs: PhysiCell (.csv) to CC3D (.pifff)



vtu

may be trickier



<https://github.com/OpenVT/playground/tree/main/PhysiCell> {pcdl2vtu.py, cb_vtu2pifff.py}

OpenVT: What might we “standardize”?

- Data formats for initial conditions (of cells and substrates)
- Concept of a cell’s phenotype?
 - May contribute to “standard” model spec
 - May contribute to digital cell lines
 - → MultiCellDS (rf. later slide)

MultiCellDS: a standard and a community for sharing multicellular data

Experiment

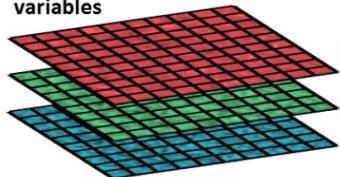
Digital Snapshot

<https://doi.org/10.1101/090696> (2016 preprint)

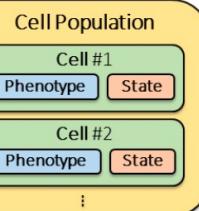
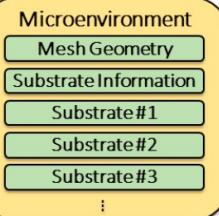
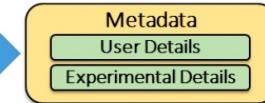
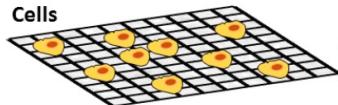
Scientist



Microenvironment variables



Cells



PhysiCell configuration files (.xml) have adopted this standard.

Core idea: have a well defined **Phenotype** for each cell type.



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

CompuCell3D
PhysiCell

PhysiCell Studio: Phenotype

<https://doi.org/10.1101/2023.10.24.563727>

<https://github.com/PhysiCell-Tools/Studio-Guide>

PhysiCell Studio: /Users/heiland/dev/PhysiCell_V1.13.1/config/cellsort_cyl1.xml

Studio File View Action Help

Config Basics Microenvironment Cell Types User Params Rules ICs Run Plot

auto number IDs when saved (beware of cells.csv using IDs)

---	Cell Type ---	-- ID --
	ctypeA	0
	ctypeB	1
	medium	2

Cycle Death Volume Mechanics Motility Secretion Interactions Intracellular Custom Data

transition rate(s) duration(s)

live cells

phase 0->0 transition rate 0.0 Fixed 1/min

XML

```
<cell_definitions>
  <cell_definition name="bacteria" ID="0">
    <phenotype>
      <cycle code="5" name="live"></cycle>
      <death></death>
      <volume></volume>
      <mechanics></mechanics>
      <motility></motility>
      <secretion></secretion>
      <cell_interactions></cell_interactions>
      <cell_transformations></cell_transformations>
    </phenotype>
```

e.g., Cell Cycle

Cycle Death Volume Mechanics Motility Secretion Interactions Intracellular

transition rate(s) duration(s)

flow cytometry separated

phase 0->1 transition rate 0.00335 Fixed
phase 1->2 transition rate 0.00208 Fixed
phase 2->3 transition rate 0.00417 Fixed
phase 3->0 transition rate 0.01667 Fixed

Phenotype: adopt in other frameworks?

PhenoCellPy: A Python package for biological cell behavior modeling

Juliano F. Gianlupi, T.J. Sego, James P. Sluka, James A. Glazier

<https://doi.org/10.1101/2023.04.12.535625>

- “...implements PhysiCell’s phenotypes and their methods in a general way, that is easily extended and embedded into other Python-based models.”
- Demo for cell cycle in CC3D (via Python interface):
<https://github.com/JulianoGianlupi/PhenoCellPy>
- Can we extend to other phenotype concepts, e.g., death, mechanics, motility?
- Can we extend to frameworks without a Python interface?

Demo: PhysiCell “biorobots” to CC3D

- <https://github.com/JulianoGianlupi/pcxml2cc3d> - I forked and edited

Given a PhysiCell model (.xml configuration file):

```
M1P~/git/pcxml2cc3d$ python convert.py
```

```
...
```

```
Generates CC3D simulation files! e.g., .cc3d, .xml, .py
```

```
<Plugin Name="Contact">
    <!-- PhysiCell doesn't have an equivalent to this plugin. Its -->
    <!-- tuning and deciding on the neighbor order is left as an -->
    <!-- exercise to the reader. -->
    <!-- A better option (to be implemented) is to use the adhesion flex -->
    <!-- Specification of adhesion energies -->
    <Energy Type1="Medium" Type2="Medium">10.0</Energy>
    <Energy Type1="Medium" Type2="WALL">5.0</Energy>
    <Energy Type1="Medium" Type2="director_cell">5.0</Energy>
    ...

```

Translate PhysiCell model to CC3D model (2)

biorobots/biorobots.cc3d - CompuCell3D Player - iteration -

Model Editor

Property	Value
> Metadata	Metadata
> Potts	Potts
> Plugin	CellType
Plugin	Volume
> Plugin	Contact
> Steppable	DiffusionSolverFE
Plugin	Secretion
Plugin	Chemotaxis
> Steppable	UniformInitializer

Cell Type Colors

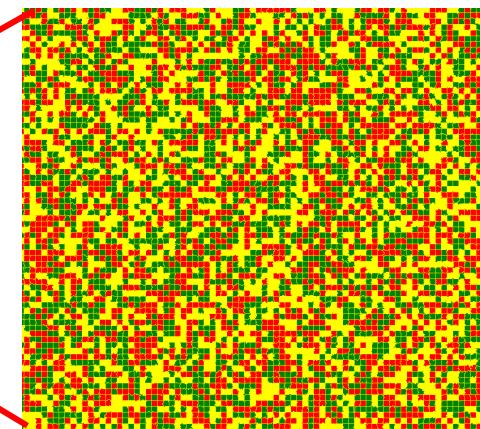
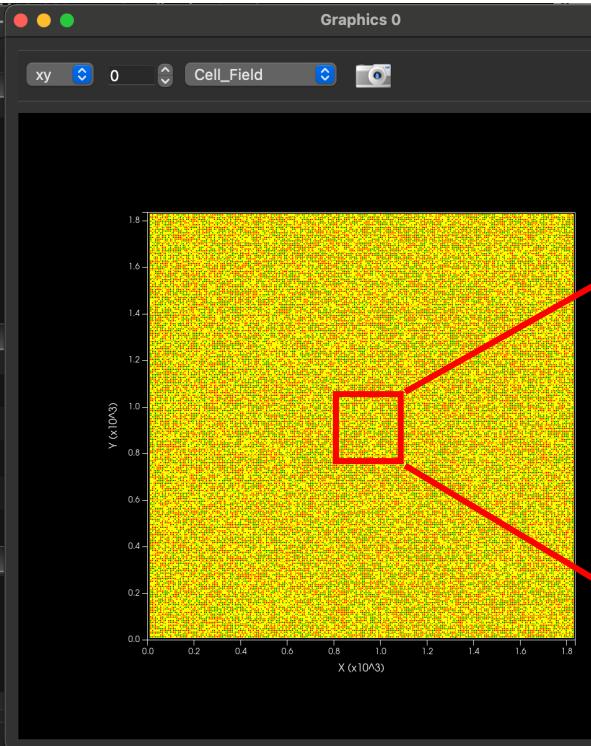
Cell Type Id	Color	Name
0	Medium	
1	WALL	<input checked="" type="checkbox"/>
2	director_cell	<input checked="" type="checkbox"/>
3	cargo_cell	<input checked="" type="checkbox"/>
4	worker_cell	<input checked="" type="checkbox"/>

Console

```
Step 0 Flips 13600/3374569 Energy -2.49196e+06 Cells 67601 Inventory=67601  
Metropolis Fast  
total number of pixel copy attempts=3374569
```

Output Errors

MC Step: 0 | COM, FPP Plugin(s) not defined - turning off Glyphs, FPP Links



The converted model actually “runs”. But we need to dig into the details.

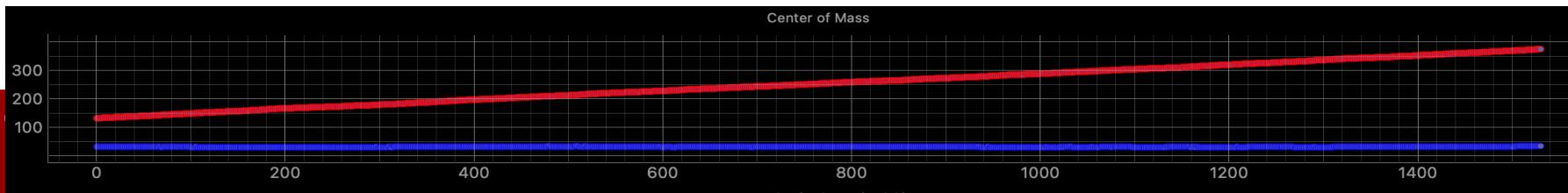
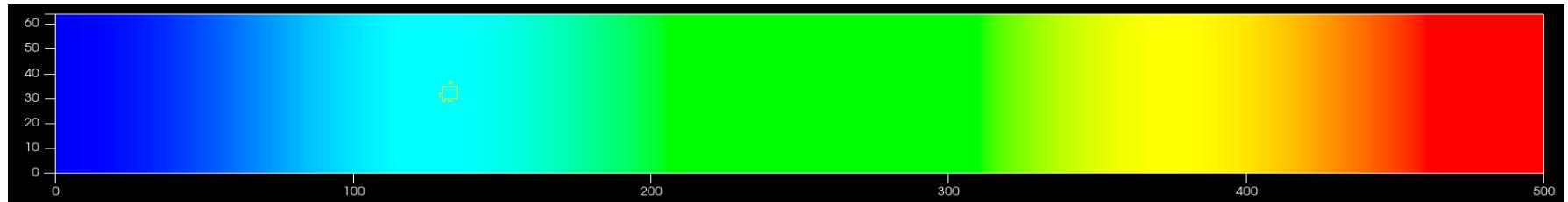
Reference models

Create a set of reference models and have each framework simulate them. Compare results:

1. Persistent Random Walk / Biased Persistent Random Walk
2. Monolayer growth and wound healing without migration
3. One cell chemotaxing across the domain
4. Cell sorting
5. Angiogenesis

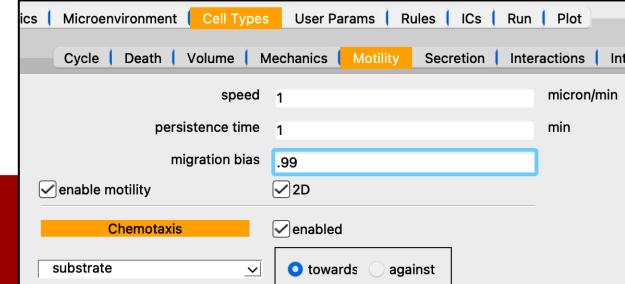
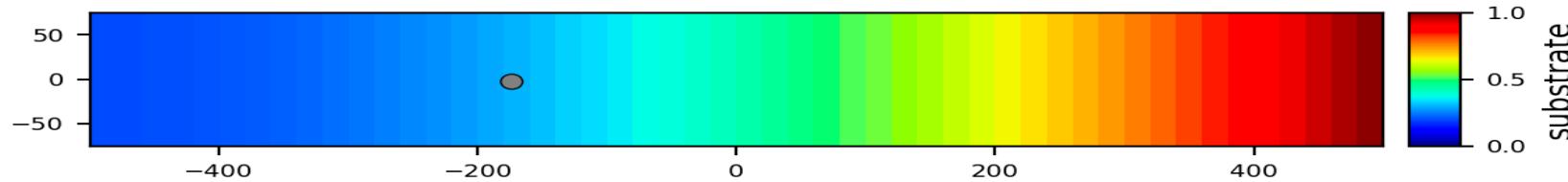
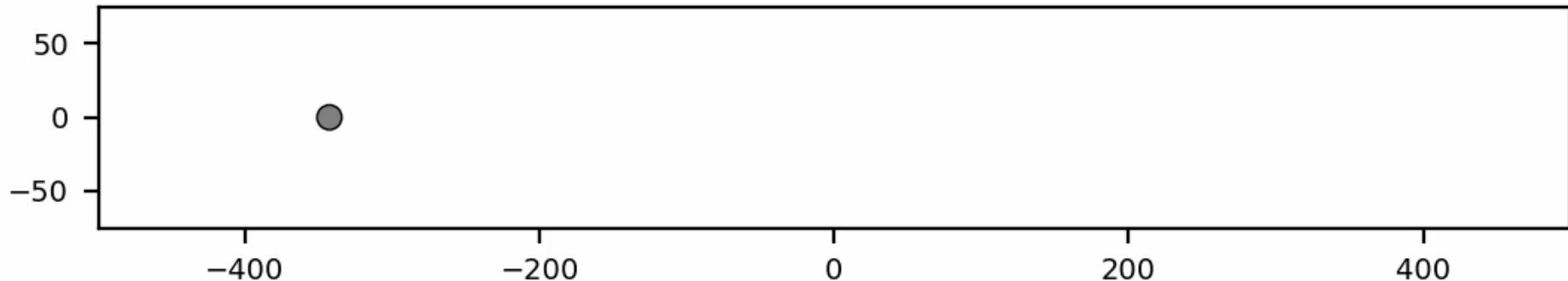
Chemotaxis, 1 cell: CC3D

- Single cell chemotaxing along a substrate gradient (thanks to Michael Getz)



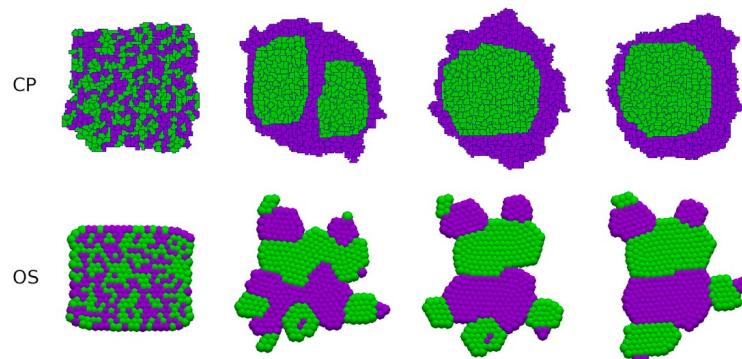
Chemotaxis, 1 cell: PhysiCell

0 days, 0 hrs, 0 mins (1 agents)



Cell sorting

Cell sorting using differential cell adhesion



Recall results from Chaste:

← Cellular Potts

← Overlapping Spheres (Center-based)

Comparing individual-based approaches to modelling the self-organization of multicellular tissues (2017). Osborne, et. al.
doi:10.1371/journal.pcbi.1005387

Integrative, modular workflows?

- Can we create workflows that allow for new functionality for frameworks? E.g., provide an anisotropic diffusion solver.
- How much work (software dev) is required?
- What is the performance hit?

Plan to experiment with:

Vivarium: an interface and engine for integrative multiscale modeling in computational biology (2022).

Agmon, et. al.. <https://doi.org/10.1093/bioinformatics/btac049>
<https://vivarium-collective.github.io/>

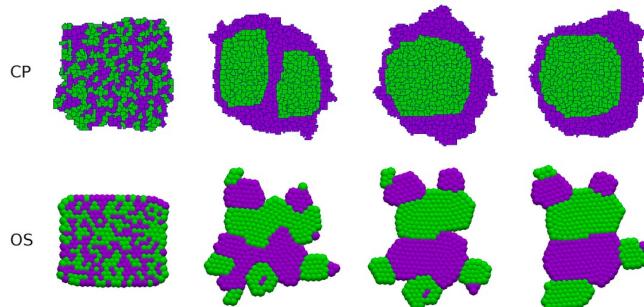
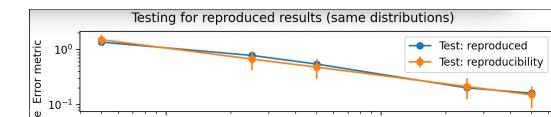
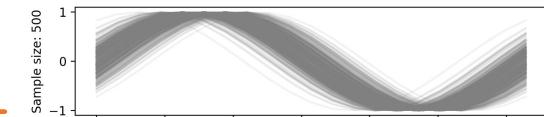
MUSCLE3: The Multiscale Coupling Library and Environment (2023). Veen, L. and Sebregts, M.

<https://doi.org/10.5281/zenodo.8396969>
<https://research-software-directory.org/software/muscle3>

IMAG WG series video: <https://www.youtube.com/watch?v=S2omygLQekk>

Reproducibility

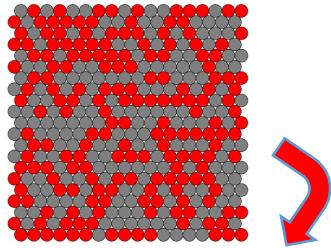
- Can we “reproduce” model results across frameworks?
- How does one quantify reproducible results?
 - Statistical measures? Geometric, Topological?
 - Snapshot of cells at “time T”? Dynamic trajectory?
 - Stochastic Simulation Reproducibility (Sego, Sheriff)
 - ◆ <https://github.com/tjsego/libSSR>



Recall cell sorting results

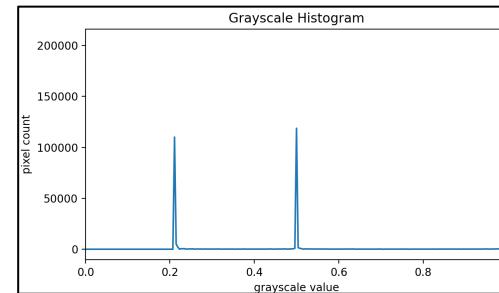
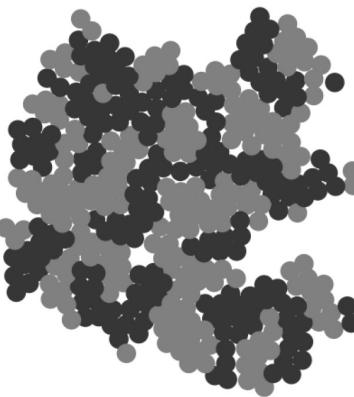
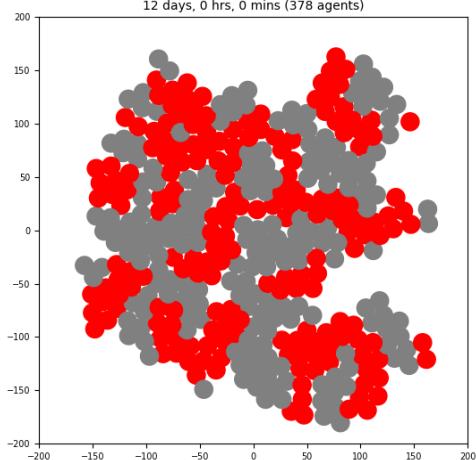
Comparing individual-based approaches to modelling the self-organization of multicellular tissues (2017).
Osborne, et. al. doi:10.1371/journal.pcbi.1005387

Reproducibility: cell clusters, image processing

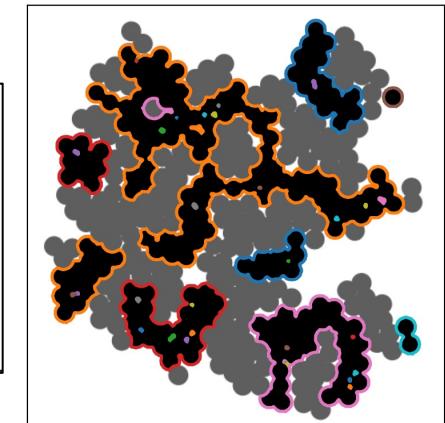


How can we quantify simulation results (to test for reproducibility)?

- e.g., cell sorting: how many clusters are there?



scikit-image: grayscale, histogram, contour to find # of clusters



Related: Jim Sluka's IMAG talk from April 18, 2024 - <https://www.youtube.com/watch?v=967zEmzm6OM>

Future Plans

- **openvt.org** – community landing site
 - Spreadsheet of frameworks' features
 - Summaries of each framework
 - Links to published models that use each framework
- Continue building bridges between frameworks:
 - Share data using agreed upon standard format
 - Share reference models' simulation results for each framework
 - Explore Phenotype concept across frameworks

Questions/Discussion

Extra slides follow

Additional related papers:

Hybrid modeling frameworks of tumor development and treatment (2019). Chamseddine and Rejniak.
DOI:10.1002/wsbm.1461

<https://github.com/OpenVT/frameworks> (cont'd)

		Lng	APIs	GUI	Web	Intracellular		Checkpointing	Operating system			Parallelism	
	Advection					Boolean	ODEs		Win	Mac	Linux	OpenMP	MPI
topic													
		C++	Python	✓	*1	✓	✓		✓	✓	✓	✓	
		C++	C++	✓	*2	✓	✓		✓	✓	✓	✓	

PhysiCell Studio: Phenotype

The screenshot shows the PhysiCell Studio interface with the following details:

- Menu Bar:** Studio, File, View, Action, Help.
- Tab Bar:** Config Basics, Microenvironment, **Cell Types**, User Params, Rules, ICs, Run, Plot.
- Left Panel:** A table for managing cell types and IDs. It includes a checkbox for "auto number IDs when saved (beware of cells.csv using IDs)".

Cell Type	ID
ctypeA	0
ctypeB	1
medium	2
- Apoptosis Tab:** Contains parameters for apoptosis and unlysed/necrotic states.

Parameter	Value	Unit
death rate	0	1/min
transition rate	radio button (disabled)	
duration	radio button (selected)	
phase 0->1 transition rate	0.001938	checkbox (Fixed) 1/min
phase 0 duration	516	checkbox (Fixed) min
unlysed fluid change rate	0.05	1/min
lysed fluid change rate	0	1/min
cytoplasmic biomass change rate	1.66667e-02	1/min
nuclear biomass change rate	5.83333e-03	1/min
calcification rate	0	1/min
relative rupture volume	2.0	
- Necrosis Tab:** Contains parameters for necrosis and unlysed/necrotic states.

Parameter	Value	Unit
death rate	0.0	1/min
transition rate	radio button (disabled)	
duration	radio button (selected)	
phase 0->1 transition rate	9.e9	checkbox (Fixed) 1/min
phase 1->2 transition rate	0.000012	checkbox (Fixed) 1/min
phase 0 duration	0	checkbox (Fixed) min
phase 1 duration	86400	checkbox (Fixed) min
unlysed fluid change rate	1.11667e-2	1/min
- Bottom Buttons:** New, Copy, Delete.

Death models:

- Apoptosis
- Necrosis

PhysiCell Studio: Phenotype

The screenshot shows the PhysiCell Studio interface with the title bar "PhysiCell Studio: /Users/heiland/dev/PhysiCell_V1.13.1/config/cellsort_cyl1.xml". The menu bar includes "Studio", "File", "View", "Action", and "Help". The top navigation bar has tabs: "Config Basics", "Microenvironment", "Cell Types" (which is highlighted in orange), "User Params", "Rules", "ICs", "Run", and "Plot". On the left, there is a sidebar with a checkbox "auto number IDs when saved (beware of cells.csv using IDs)" and a table for managing cell types:

Cell Type	ID
ctypeA	0
ctypeB	1
medium	2

The main content area is titled "Volume" and contains the following parameters:

total	2494	micron ³
fluid fraction	0.75	
nuclear	540	micron ³
fluid change rate	0.05	1/min
cytoplasmic biomass change rate	0.0045	1/min
nuclear biomass change rate	0.0055	1/min
calcification fraction	0	
calcified rate	0	1/min
relative rupture volume	2.0	

Volume

PhysiCell Studio: Phenotype

The screenshot shows the PhysiCell Studio interface with the 'Cell Types' tab selected. On the left, there is a table for managing cell types and IDs:

Cell Type	ID
ctypeA	0
ctypeB	1
medium	2

On the right, the 'Mechanics' tab is active, displaying various parameters:

- cell-cell adhesion strength: 0.4 micron/min
- cell-cell repulsion strength: 10.0 micron/min
- relative max adhesion distance: 1.25
- cell adhesion affinity: ctypeA (dropdown) 1.0
- Options:
 - relative equilibrium distance: 1.8 (checkbox: enable)
 - absolute equilibrium distance: 15.12 (checkbox: enable) micron
 - elastic constant: 0.01 1/min
 - attachment rate: 0 1/min
 - detachment rate: 0.0 1/min

Mechanics: adhesion, repulsion (potential fns); spring mech.

PhysiCell Studio: Phenotype

The screenshot shows the PhysiCell Studio interface with the title "PhysiCell Studio: /Users/heiland/dev/PhysiCell_V1.13.1/config/cellsort_cyl1.xml". The menu bar includes Studio, File, View, Action, and Help. The top navigation bar has tabs: Config Basics, Microenvironment, Cell Types (highlighted in orange), User Params, Rules, ICs, Run, and Plot.

The "Cell Types" tab is active, showing a table of cell types and their IDs:

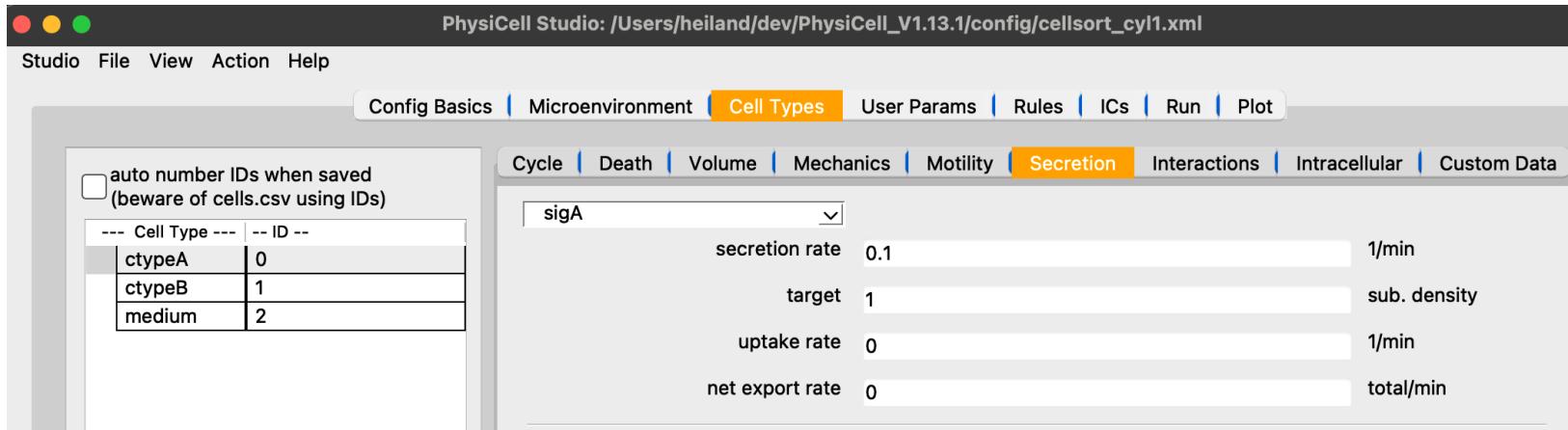
Cell Type	ID
ctypeA	0
ctypeB	1
medium	2

The "Motility" tab is selected under the Cell Types tab. The configuration for cell type ctypeA is as follows:

- speed: 1.0 micron/min
- persistence time: 1 min
- migration bias: 0.0
- enable motility
- 2D
- Chemotaxis** is enabled (checkbox checked).
 - sigA dropdown menu
 - towards/against radio buttons (towards is selected)
- Advanced Chemotaxis** is disabled (checkbox unchecked).
 - sigA dropdown menu
 - sensitivity: 0.0
 - normalize gradient checkbox (unchecked)

Motility: basic motility; chemotaxis

PhysiCell Studio: Phenotype



Secretion: of selected substrates

PhysiCell Studio: Phenotype

The screenshot shows the PhysiCell Studio interface. The title bar reads "PhysiCell Studio: /Users/heiland/dev/PhysiCell_V1.13.1/config/cellsort_cyl1.xml". The menu bar includes "Studio", "File", "View", "Action", and "Help". The top navigation bar has tabs: "Config Basics", "Microenvironment", "Cell Types" (which is selected and highlighted in orange), "User Params", "Rules", "ICs", "Run", and "Plot". Below this, a secondary navigation bar has tabs: "Cycle", "Death", "Volume", "Mechanics", "Motility", "Secretion", "Interactions" (selected and highlighted in orange), "Intracellular", and "Custom Data". On the left, there is a configuration panel with a checkbox for "auto number IDs when saved (beware of cells.csv using IDs)" and a table for mapping cell types to IDs:

Cell Type	ID
ctypeA	0
ctypeB	1
medium	2

The main workspace displays interaction parameters for cell type ctypeA:

- dead phagocytosis rate: 0 1/min
- live phagocytosis rate: ctypeA 0.0 1/min
- attack rate: ctypeA 0.0 1/min
- damage rate: 1 1/min
- fusion rate: ctypeA 0.0 1/min
- transformation rate: ctypeA 0.0 1/min

Interactions: how does one cell type interact with others