Slides, videos, links and more:
https://github.com/physicell-training/ws2021

Preparing your PhysiCell model to share on nanoHUB

Randy Heiland
@PhysiCell

PhysiCell Project

July 25-31, 2021

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project
PhysiCell.org
@PhysiCell

Overview

- nanoHUB (https://nanohub.org/) is an open and free platform for computational research, education, and collaboration in nanotechnology, materials science, and related fields.

- nanoHUB lets users run interactive apps from a browser.

- This tutorial will explain how you port an existing PhysiCell 2-D model to run on nanoHUB.

Assumptions

- You have installed the Anaconda Python distribution:
  - https://github.com/MathCancer/PhysiCell/blob/master/documentation/Quickstart.md#python
  - If you need more setup details: https://github.com/physicell-training/ws2021#pre-workshop-materials

- You have a working PhysiCell 2-D model that adheres to the default directory structure, file naming scheme (`Makefile`, `main.cpp`, `config/PhysiCell_settings.xml`), and contains the `output/initial.xml` file from a simulation.

- You have downloaded and installed (unzipped) the latest release from https://github.com/PhysiCell-Tools/PhysiCell-Jupyter-GUI/releases

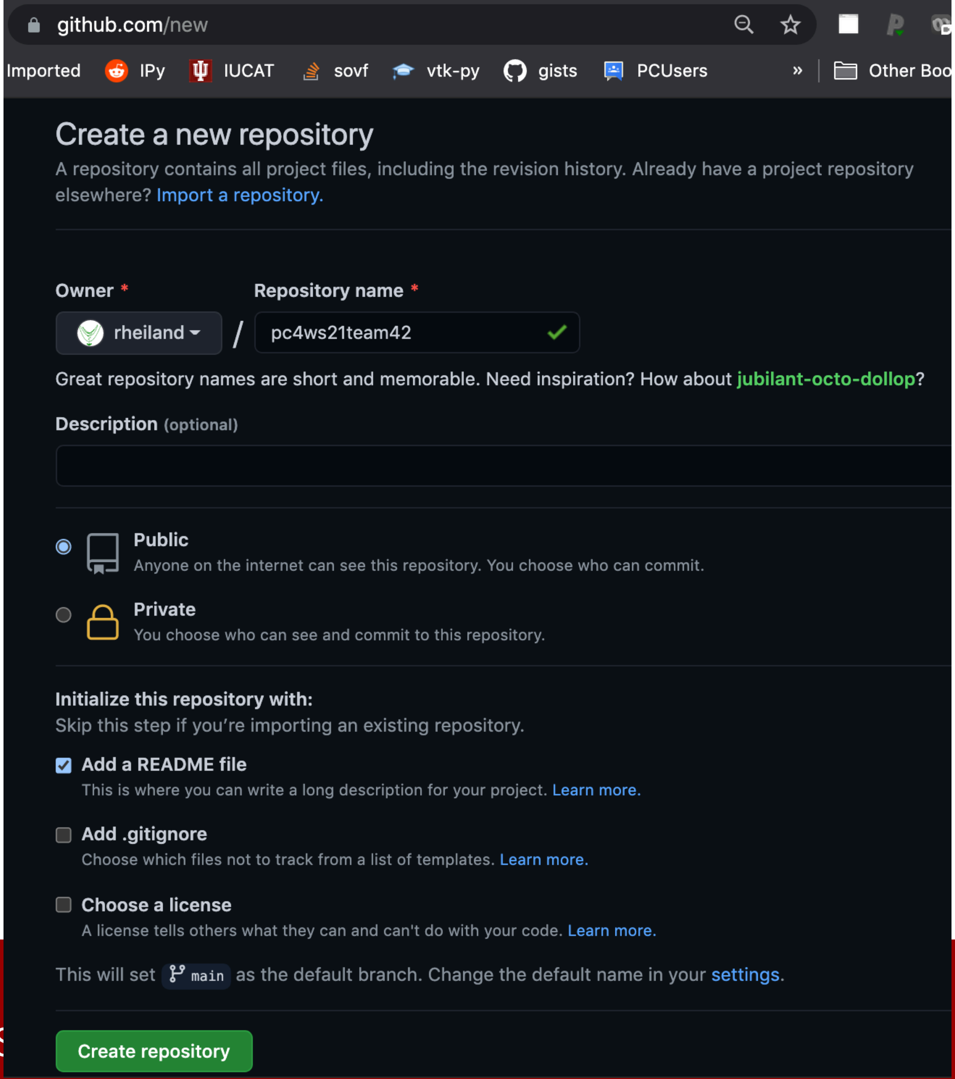
Detailed instructions

- Follow the steps here:
  - https://github.com/PhysiCell-Tools/PhysiCell-Jupyter-GUI/blob/master/README.md
  - https://github.com/PhysiCell-Tools/PhysiCell-Jupyter-GUI/blob/master/README.md#steps-to-follow
    - ♦ Instead of using "ise_proj1", you may want to use: "pc4ws21teamN" (PhysiCell for Workshop 2021, where N=1,2,etc (whatever your team # is)) – see next slide

The following slides try to illustrate these steps.

  - These instructions should let you create a Jupyter notebook that you can run locally (on your computer) before installing it on nanoHUB.
  - Creating a nanoHUB app is considered optional (https://github.com/PhysiCell-Tools/PhysiCell-Jupyter-GUI/blob/master/README.md#create-a-nanohub-tool-optional), but we are assuming you will do it in this tutorial. Let us know if you need help.

Create a new GitHub repo for your nanoHUB project. E.g., I created “pc4ws21team42”


github.com/new

Imported IPy IUCAT sovf vtk-py gists PCUsers » Other Boo

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * rheiland / Repository name * pc4ws21team42

Great repository names are short and memorable. Need inspiration? How about jubilant-octo-dollop?

Description (optional)

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you’re importing an existing repository.

[✓] Add a README file
This is where you can write a long description for your project. Learn more.

[ ] Add .gitignore
Choose which files not to track from a list of templates. Learn more.

[ ] Choose a license
A license tells others what they can and can't do with your code. Learn more.

This will set `main` as the default branch. Change the default name in your settings.

Create repository

Using biorobots as an example (1)

```
~$ unzip PhysiCell_V.1.9.0.zip
...
~$ mv PhysiCell  PhysiCell_v1.9.0
~$ cd PhysiCell_v1.9.0
~/PhysiCell_v1.9.0$ make biorobots-sample
cp ./sample_projects/biorobots/custom_modules/* ./custom_modules/
touch main.cpp && cp main.cpp main-backup.cpp
cp ./sample_projects/biorobots/main-biorobots.cpp ./main.cpp
cp Makefile Makefile-backup
cp ./sample_projects/biorobots/Makefile .
cp ./config/PhysiCell_settings.xml ./config/PhysiCell_settings-backup.xml
cp ./sample_projects/biorobots/config/* ./config/

~/PhysiCell_v1.9.0$ make
...
~/PhysiCell_v1.9.0$ biorobots
... Let it run just for a very brief time (to generate some initial output) and then ctl-c to kill it
```

Using biorobots as an example (2)

- Verify you have these 4 critical files:

```
~/PhysiCell_v1.9.0$ ls Makefile main.cpp config/PhysiCell_settings.xml output/initial.xml
Makefile                                main.cpp
config/PhysiCell_settings.xml           output/initial.xml
```

Clone your GitHub repo

```
~$ git clone git@github.com:rheiland/pc4ws21team42.git
Cloning into 'pc4ws21team42'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
~$

~/PhysiCell-Jupyter-GUI-1.4$ python setup_new_proj.py  ~/pc4ws21team42  ~/PhysiCell_v1.9.0  pc4ws21team42
...
```

This will generate these files in your cloned nanoHUB project:

```
~/pc4ws21team42$ ls
LICENSE          examples/          rappture/
README.md        make_my_tool.py    src/
bin/             middleware/        tmpdir/
data/            mod_makefile.py
doc/             pc4ws21team42.ipynb    ← This will be your project's name
```

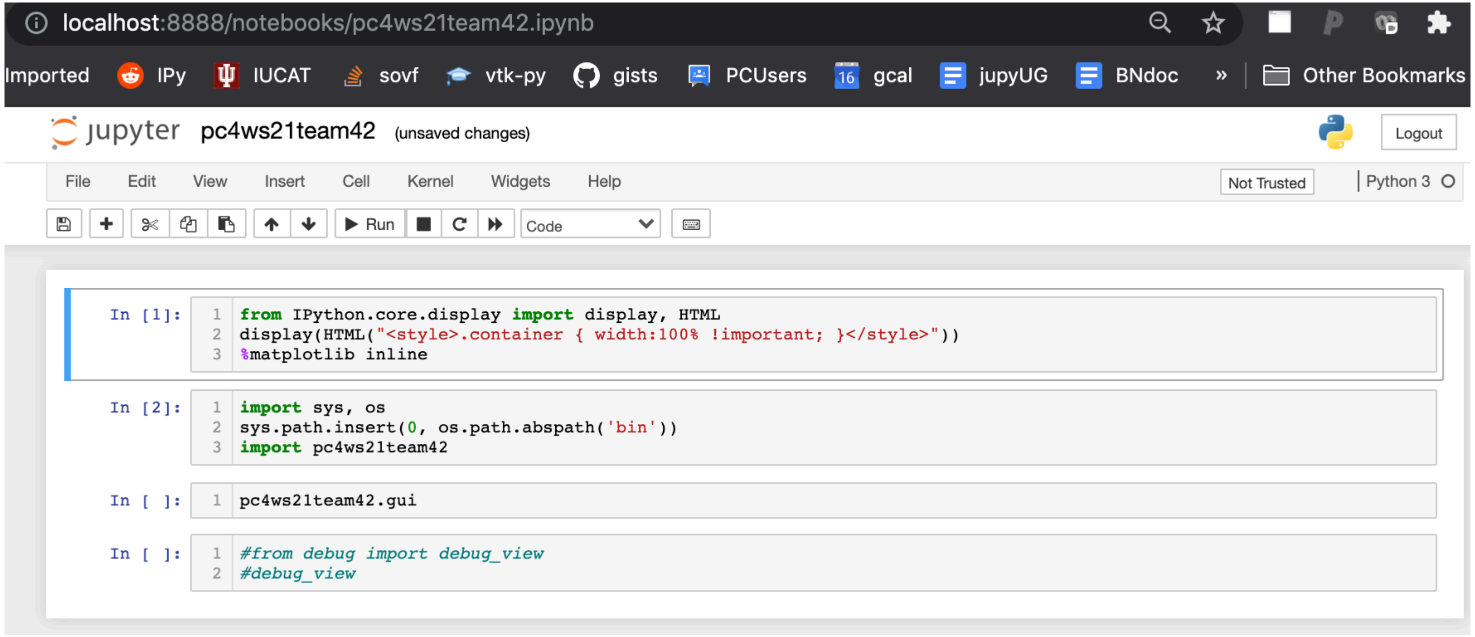
Test your notebook on your computer

Have a Web browser running, then run the following command from a shell:

~/pc4ws21team42$ **jupyter notebook pc4ws21team42.ipynb**

...

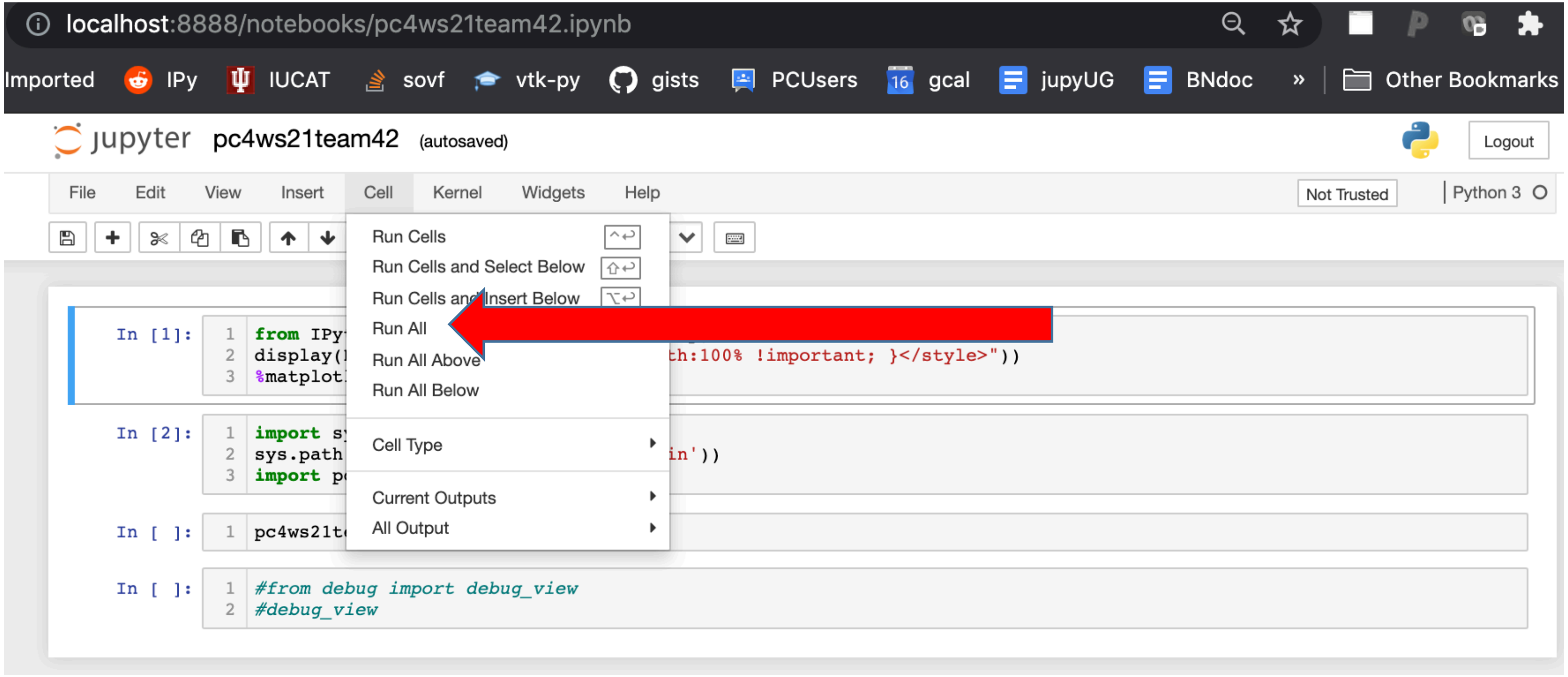
You should see this Jupyter notebook displayed in your browser:



localhost:8888/notebooks/pc4ws21team42.ipynb
Imported IPy IUCAT sovf vtk-py gists PCUsers gcal jupyUG BNdoc » Other Bookmarks
jupyter pc4ws21team42 (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code

```
In [1]: from IPython.core.display import display, HTML
        display(HTML("<style>.container { width:100% !important; }</style>"))
        %matplotlib inline

In [2]: import sys, os
        sys.path.insert(0, os.path.abspath('bin'))
        import pc4ws21team42

In [ ]: pc4ws21team42.gui

In [ ]: #from debug import debug_view
        #debug_view
```

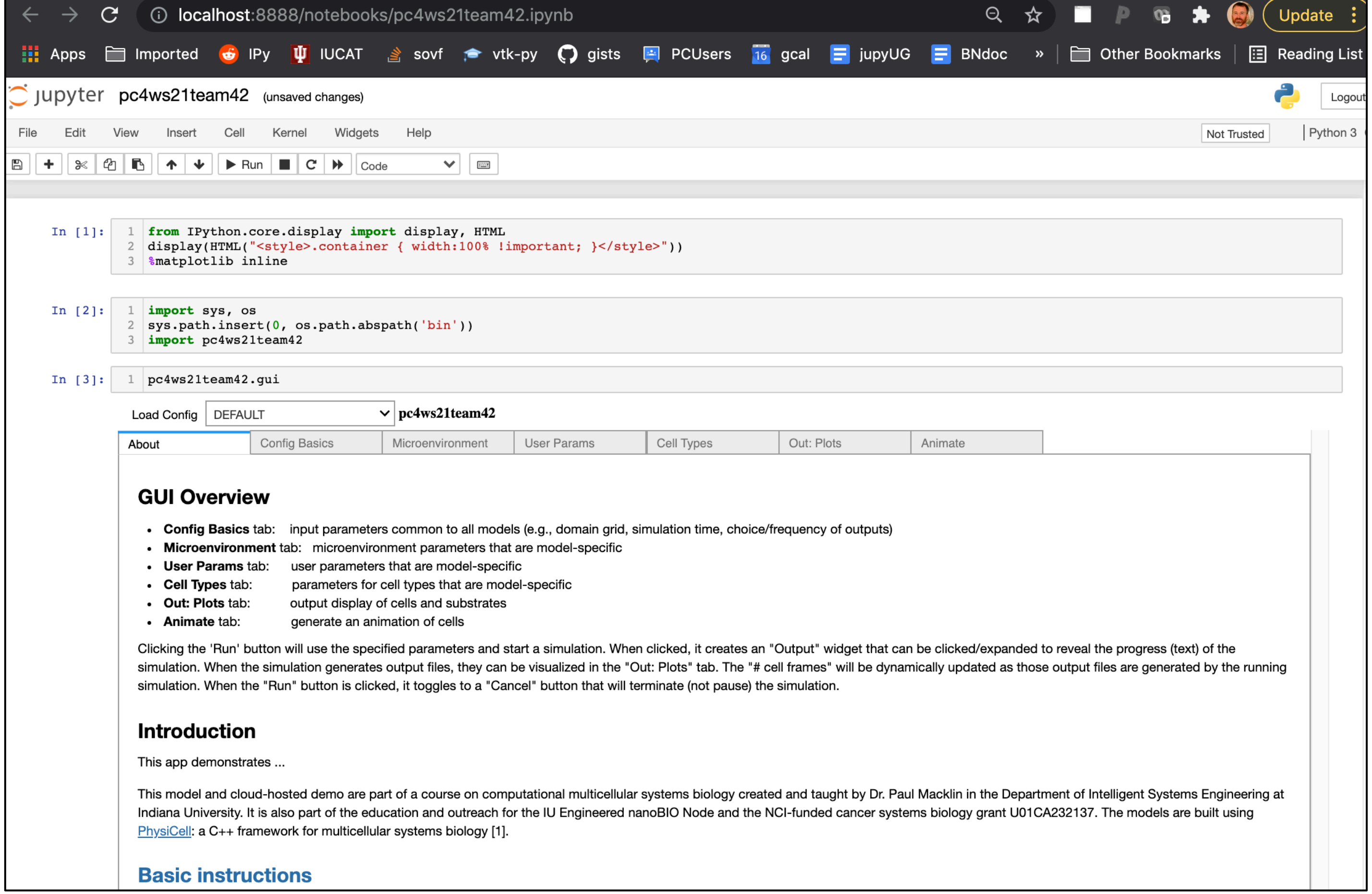
Click the ‘Cell’ menu, ‘Run All’, wait for each notebook “cell” to execute...



```
In [1]:  from IPy
         display(
         %matplot
                                          th:100% !important; }</style>"))
In [2]:  import s
         sys.path                         in'))
         import p
In [ ]:  pc4ws21t
In [ ]:  #from debug import debug_view
         #debug_view
```

You should see a similar looking notebook for your project.

Now or later:
You can leave this running, then edit the doc/about.html file to update this page's contents and (see next page).



```
In [1]: from IPython.core.display import display, HTML
        display(HTML("<style>.container { width:100% !important; }</style>"))
        %matplotlib inline

In [2]: import sys, os
        sys.path.insert(0, os.path.abspath('bin'))
        import pc4ws21team42

In [3]: pc4ws21team42.gui
```

Load Config: DEFAULT — pc4ws21team42

About | Config Basics | Microenvironment | User Params | Cell Types | Out: Plots | Animate

GUI Overview

- **Config Basics** tab: input parameters common to all models (e.g., domain grid, simulation time, choice/frequency of outputs)
- **Microenvironment** tab: microenvironment parameters that are model-specific
- **User Params** tab: user parameters that are model-specific
- **Cell Types** tab: parameters for cell types that are model-specific
- **Out: Plots** tab: output display of cells and substrates
- **Animate** tab: generate an animation of cells

Clicking the 'Run' button will use the specified parameters and start a simulation. When clicked, it creates an "Output" widget that can be clicked/expanded to reveal the progress (text) of the simulation. When the simulation generates output files, they can be visualized in the "Out: Plots" tab. The "# cell frames" will be dynamically updated as those output files are generated by the running simulation. When the "Run" button is clicked, it toggles to a "Cancel" button that will terminate (not pause) the simulation.

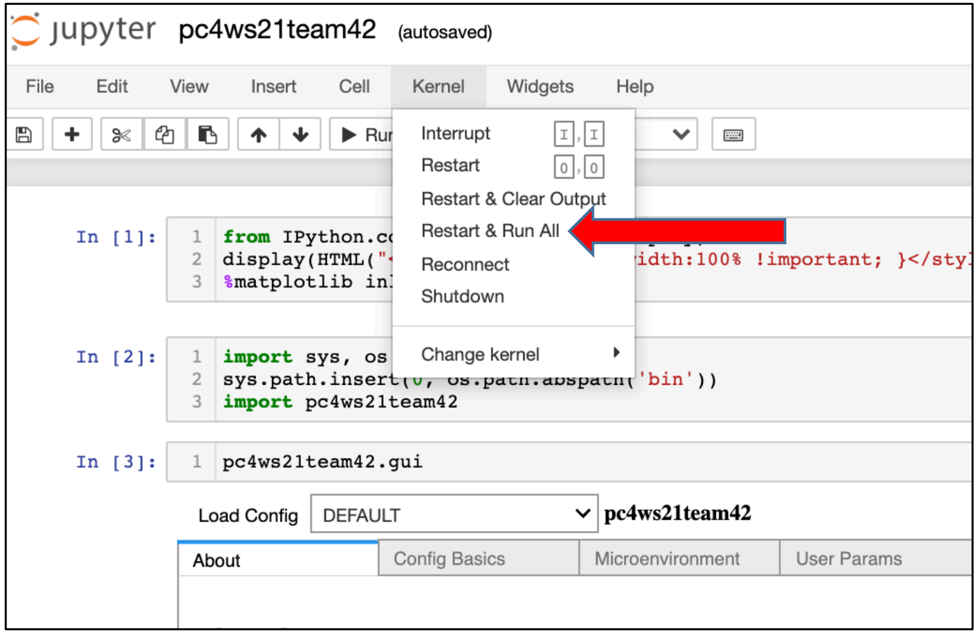
Introduction

This app demonstrates ...

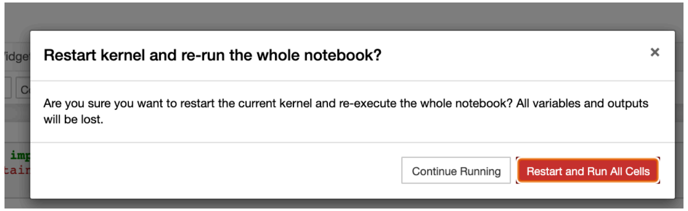
This model and cloud-hosted demo are part of a course on computational multicellular systems biology created and taught by Dr. Paul Macklin in the Department of Intelligent Systems Engineering at Indiana University. It is also part of the education and outreach for the IU Engineered nanoBIO Node and the NCI-funded cancer systems biology grant U01CA232137. The models are built using PhysiCell: a C++ framework for multicellular systems biology [1].

Basic instructions

After editing the "doc/about.html", Click the 'Kernel' menu, 'Restart & Run All', then click the pop-up button to confirm:

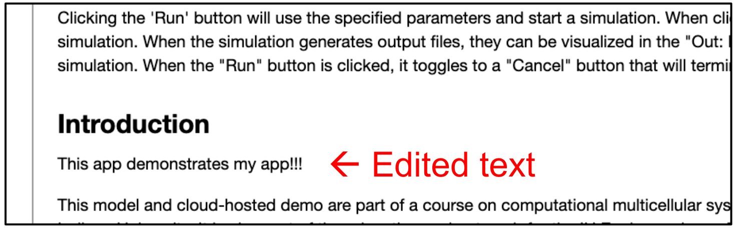

jupyter pc4ws21team42 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Interrupt
Restart
Restart & Clear Output
Restart & Run All
Reconnect
Shutdown
Change kernel
Load Config DEFAULT pc4ws21team42
About Config Basics Microenvironment User Params

```
In [1]: from IPython.co
        display(HTML("<                    idth:100% !important; }</sty
        %matplotlib inl

In [2]: import sys, os
        sys.path.insert(0, os.path.abspath('bin'))
        import pc4ws21team42

In [3]: pc4ws21team42.gui
```


Restart kernel and re-run the whole notebook?

Are you sure you want to restart the current kernel and re-execute the whole notebook? All variables and outputs will be lost.

The notebook will refresh, showing your edited text:

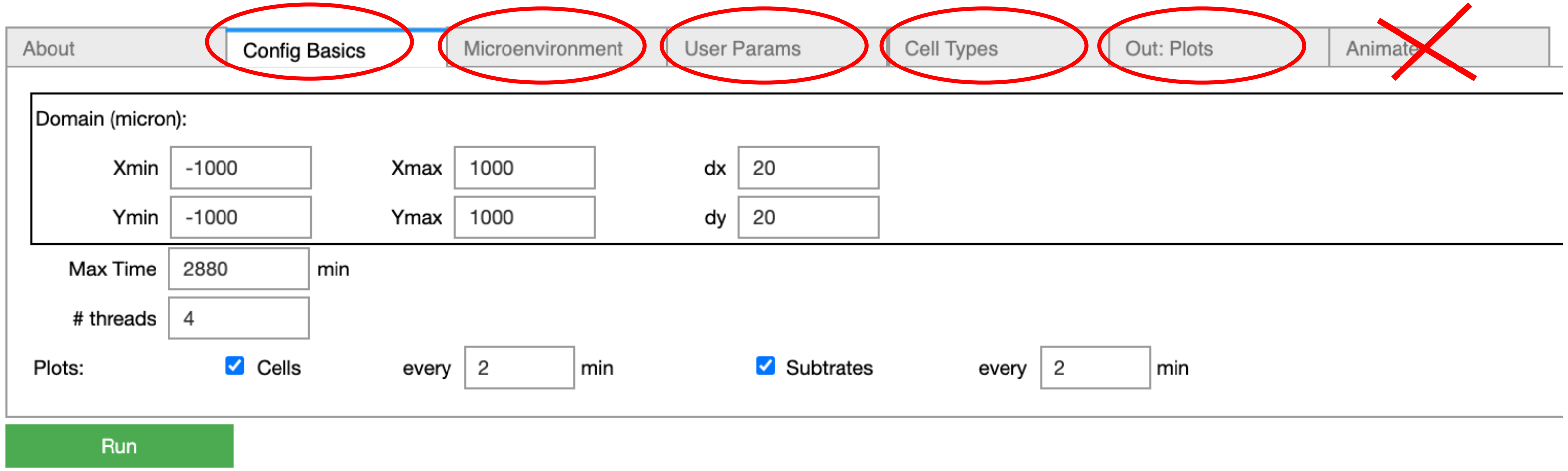

Clicking the 'Run' button will use the specified parameters and start a simulation. When cli
simulation. When the simulation generates output files, they can be visualized in the "Out:
simulation. When the "Run" button is clicked, it toggles to a "Cancel" button that will termi

Introduction

This app demonstrates my app!!! ← Edited text

This model and cloud-hosted demo are part of a course on computational multicellular sys

Explore the other tabs of the GUI (but disregard the Animate tab)



Domain (micron):

Xmin	-1000	Xmax	1000	dx	20
Ymin	-1000	Ymax	1000	dy	20

Max Time 2880 min

threads 4

Plots: ✓ Cells every 2 min ✓ Subtrates every 2 min

Final steps for nanoHUB app

- If your local Jupyter notebook looks correct, you have the option of compiling your project and making sure you can Run it from the notebook - rf. Steps 4 and 5 in:
- https://github.com/PhysiCell-Tools/PhysiCell-Jupyter-GUI/blob/master/README.md#steps-to-follow

- However, you can probably skip straight to getting it ready to test on nanoHUB:
- https://github.com/PhysiCell-Tools/PhysiCell-Jupyter-GUI/blob/master/README.md#create-a-nanohub-tool-optional
  (not "optional" if you plan to do it ☺)

Support

- We encourage you to join and actively use the PhysiCell community Slack channel. There, you can post questions (#troubleshooting), answer questions, and (hopefully) share successful modeling stories.
- Alternatively, you can submit problem tickets at https://sourceforge.net/p/physicell/tickets/
- Finally, please follow us on Twitter @PhysiCell and @MathCancer.

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)

Training Materials:

- Administrative supplement to NCI U01CA232137 (Year 2)