Slides, videos, links and more:
https://github.com/physicell-training/ws2023

Lesson 7: Working with PhysiCell Data in Python

Elmar Bucher
@MathCancer

PhysiCell Project

August 2023

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

PhysiCell Project
PhysiCell.org
@PhysiCell

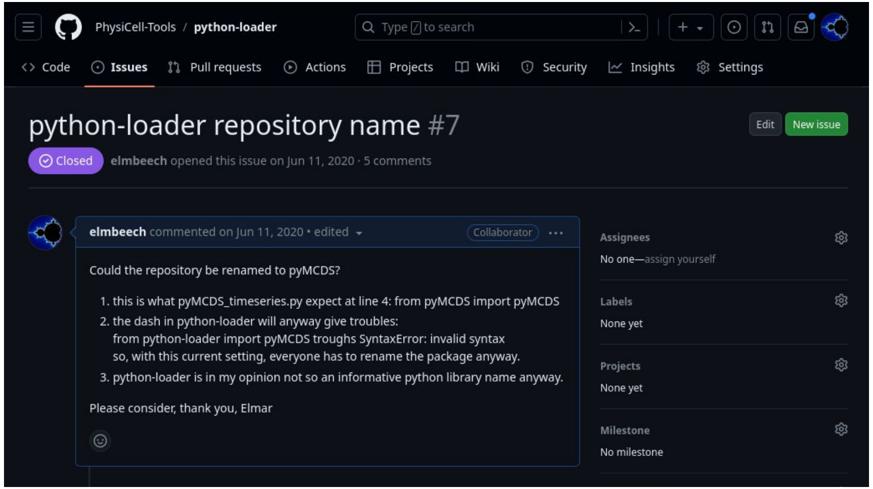
History

- 2019-09-28: **first release** Physicell-Tools/python-loader.

1. copy `pyMCDS.py` into your output folder.
2. `from pyMCDS import pyMCDS`

- it is still possible to run pcdl like this!

- an issue that bothered me …



python-loader repository name #7

Closed elmbeech opened this issue on Jun 11, 2020 · 5 comments

elmbeech commented on Jun 11, 2020 • edited

Could the repository be renamed to pyMCDS?

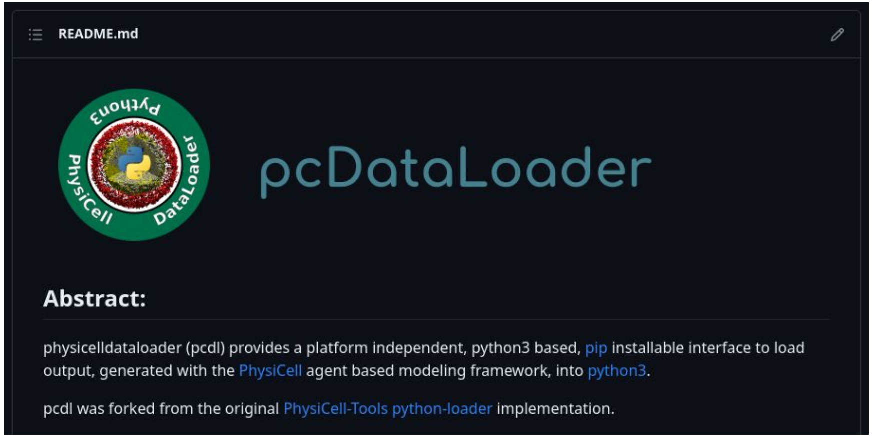
1. this is what pyMCDS_timeseries.py expect at line 4: from pyMCDS import pyMCDS
2. the dash in python-loader will anyway give troubles:
   from python-loader import pyMCDS troughs SyntaxError: invalid syntax
   so, with this current setting, everyone has to rename the package anyway.
3. python-loader is in my opinion not so an informative python library name anyway.

Please consider, thank you, Elmar

https://github.com/PhysiCell-Tools/python-loader

History

- 2022-08: I joined the lab and overtook **maintenance** of the phython-loder.
- physicell data loader was **forked**.
- **aim**: …



Abstract:

physicelldataloader (pcdl) provides a platform independent, python3 based, pip installable interface to load output, generated with the PhysiCell agent based modeling framework, into python3.

pcdl was forked from the original PhysiCell-Tools python-loader implementation.

pcdl - “PhysiCell’s USB port”

Focus:

- stay **backwards compatible**, as much as possible.
- stay platform independent (**Windows**, **MacOS**, **Linux**).
- be **pip** installable.
- load **PhysiCell** output into **Python3**.
- supply the loaded data in well established, standardized data formats.
  - pandas **DataFrame** and **Series**
  - scverse **AnnData** objects

In other words:

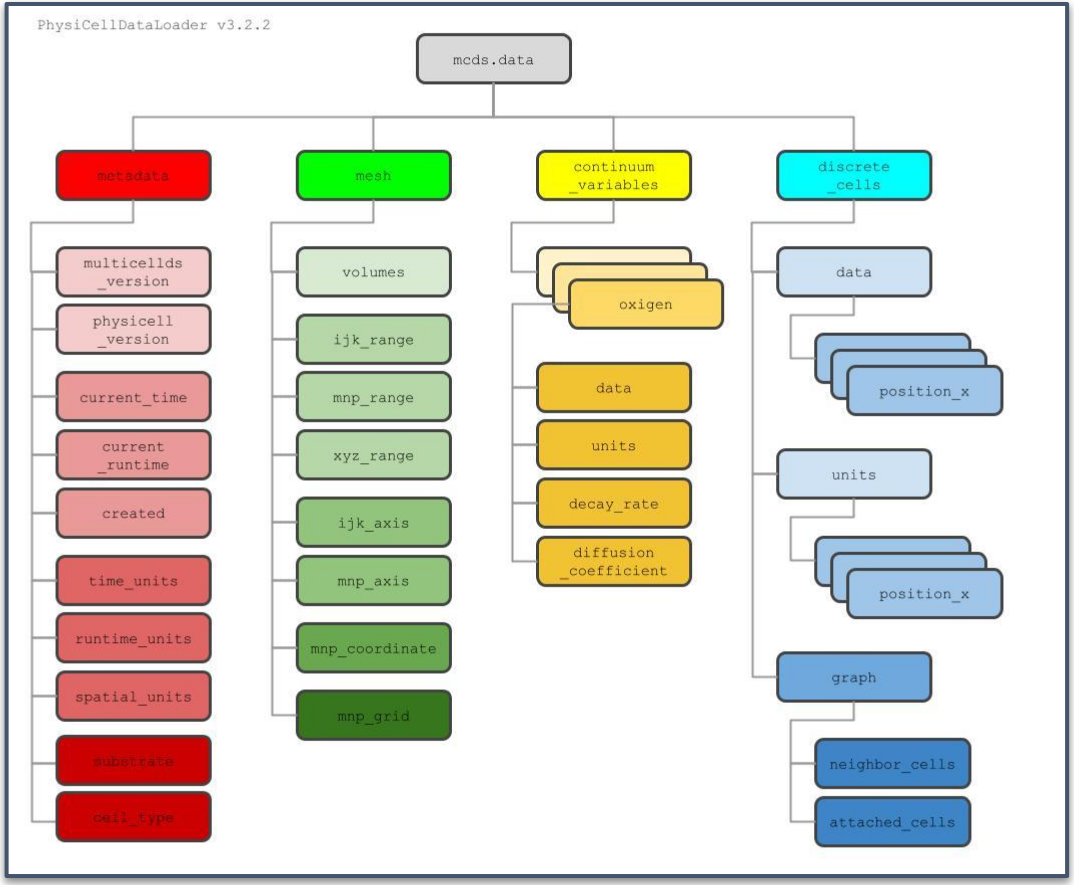
- pcdl **does not try to be** analysis software.
- pcdl **tries to be the** default connector between PhysiCell output and analysis software.

In this presentation, we will see, how far we come with this strategy.

Blueprint

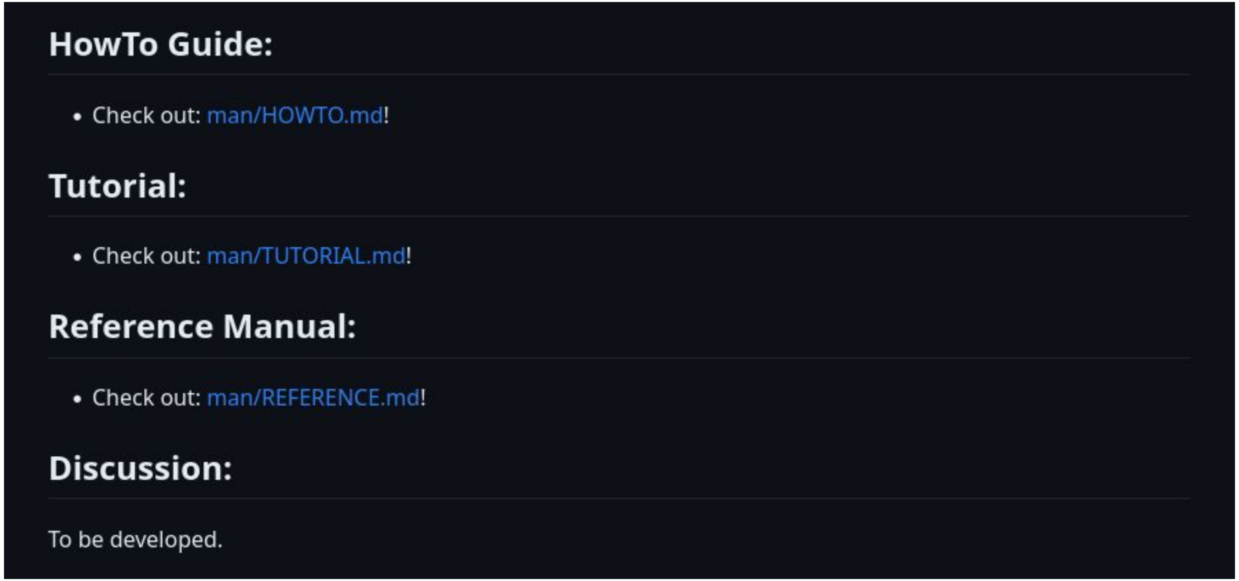
one **time step** =
one **mcds object** =
one Python3 **dictionary**
and a set of **functions** to
access the data.

mcds: **Multi Cellular Data Standard**



pcdl - all the details

- https://github.com/elmbeech/physicelldataloader/blob/master/README.md



HowTo Guide:

- Check out: man/HOWTO.md!

Tutorial:

- Check out: man/TUTORIAL.md!

Reference Manual:

- Check out: man/REFERENCE.md!

Discussion:

To be developed.

pcdl[all] - python library dependencies

pip install pcdl # light weight, the basics:

- matplotlib
- numpy
- pandas
- scipy

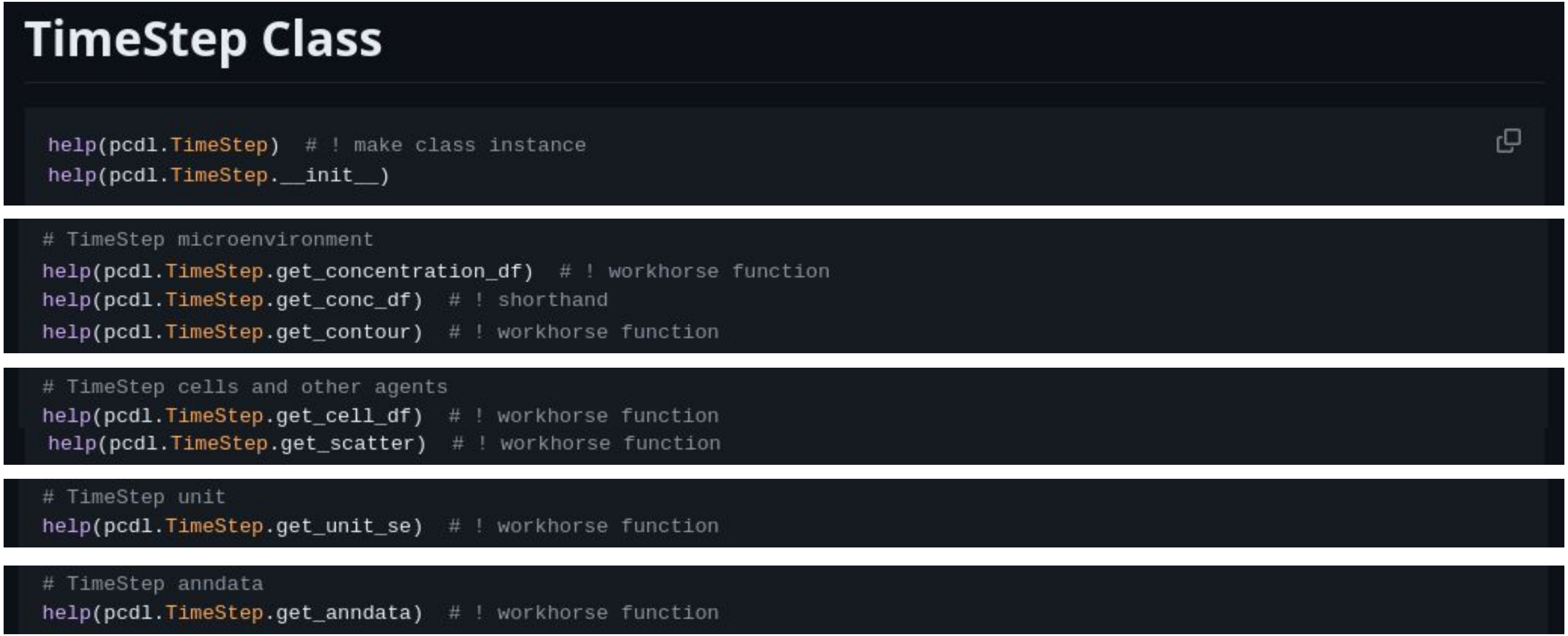
pip install pcdl[scverse] # heavy weight:

- anndata - for single cell data analysis

pip install pcdl[data] # used for the tutorial:

- requests - for http downloading the data files

pcdl - the workhorse functions for data analysis



TimeStep Class

```
help(pcdl.TimeStep)  # ! make class instance
help(pcdl.TimeStep.__init__)
```

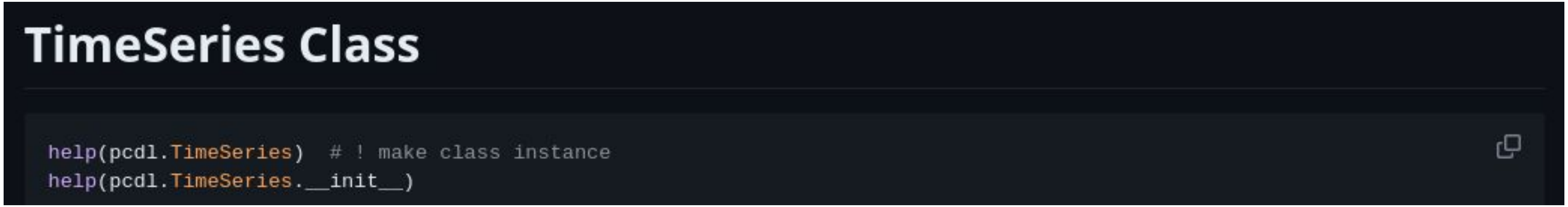
```
# TimeStep microenvironment
help(pcdl.TimeStep.get_concentration_df)  # ! workhorse function
help(pcdl.TimeStep.get_conc_df)  # ! shorthand
help(pcdl.TimeStep.get_contour)  # ! workhorse function
```

```
# TimeStep cells and other agents
help(pcdl.TimeStep.get_cell_df)  # ! workhorse function
help(pcdl.TimeStep.get_scatter)  # ! workhorse function
```

```
# TimeStep unit
help(pcdl.TimeStep.get_unit_se)  # ! workhorse function
```

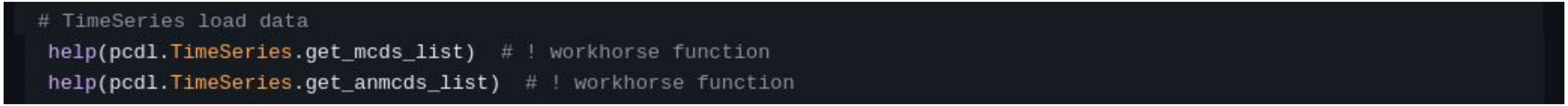
```
# TimeStep anndata
help(pcdl.TimeStep.get_anndata)  # ! workhorse function
```

pcdl - the workhorse functions for data analysis

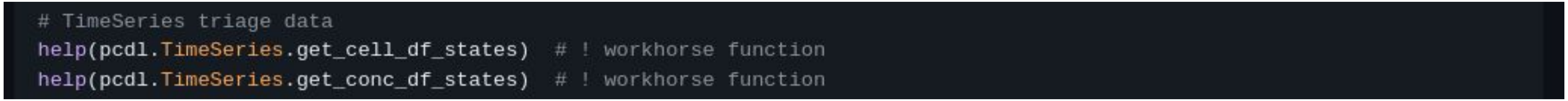


TimeSeries Class

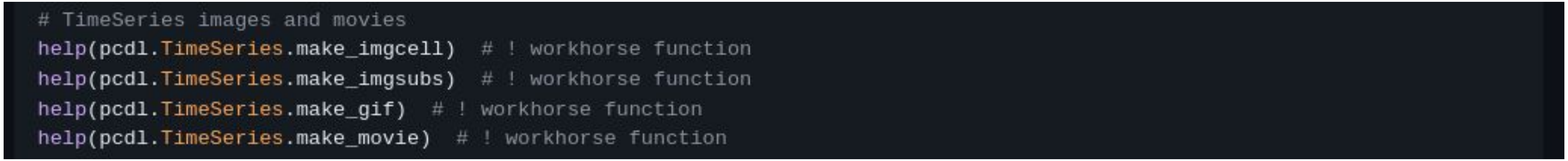
```
help(pcdl.TimeSeries)  # ! make class instance
help(pcdl.TimeSeries.__init__)
```



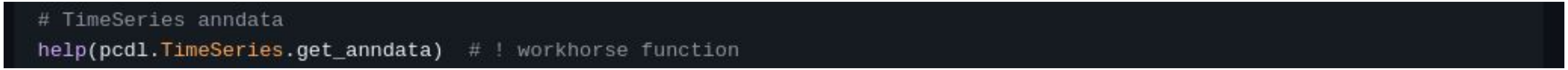
```
# TimeSeries load data
 help(pcdl.TimeSeries.get_mcds_list)  # ! workhorse function
 help(pcdl.TimeSeries.get_anmcds_list)  # ! workhorse function
```



```
# TimeSeries triage data
help(pcdl.TimeSeries.get_cell_df_states)  # ! workhorse function
help(pcdl.TimeSeries.get_conc_df_states)  # ! workhorse function
```



```
# TimeSeries images and movies
help(pcdl.TimeSeries.make_imgcell)  # ! workhorse function
help(pcdl.TimeSeries.make_imgsubs)  # ! workhorse function
help(pcdl.TimeSeries.make_gif)  # ! workhorse function
help(pcdl.TimeSeries.make_movie)  # ! workhorse function
```



```
# TimeSeries anndata
help(pcdl.TimeSeries.get_anndata)  # ! workhorse function
```

Data analysis with pcdl …

- is not **source code** programming!
  (Although, when you write an analysis tool, you should use pcdl to load PhysiCell output.)

- is interactive “**REPL** (read-eval-print loop) programming”.
  REPL programming’s basis is “one-liner” code.
  REPL programming is ideal for **ad hoc, on the fly data analysis**.
  REPL programming is more agile than any **GUI** (graphical user interface) can be.

In the beginning, I will move slowly, to show how I do REPL programming.
Afterward I will move fast, to give an overview about what is possible.

live demo!

pcdl_ws2023.ipynb

- https://github.com/elmbeech/physicelldataloader/blob/master/man/jupyter/pcdl_ws2023.ipynb

Conclusion

- pcdl is simply and **interface**.
- don’t re-invent the wheel: it is a fruitful approach to learn, make use of, and contribute to in the filed **well established high-level Python3 libraries**.

Where to go from here?

the basics:

- learn core python3: https://www.python.org/ [resources: https://nostarch.com/python-kids-2nd-edition, https://realpython.com/]
- learn numpy: https://numpy.org/
- learn scipy: https://scipy.org/
- learn pandas: https://pandas.pydata.org
- learn matplotlib: https://matplotlib.org/

depending on your needs, learn:

- statsmodels: https://www.statsmodels.org/stable/index.html
- sklearn: https://scikit-learn.org/stable/
- pytorch: https://pytorch.org/
- skimage: https://scikit-image.org/
- …

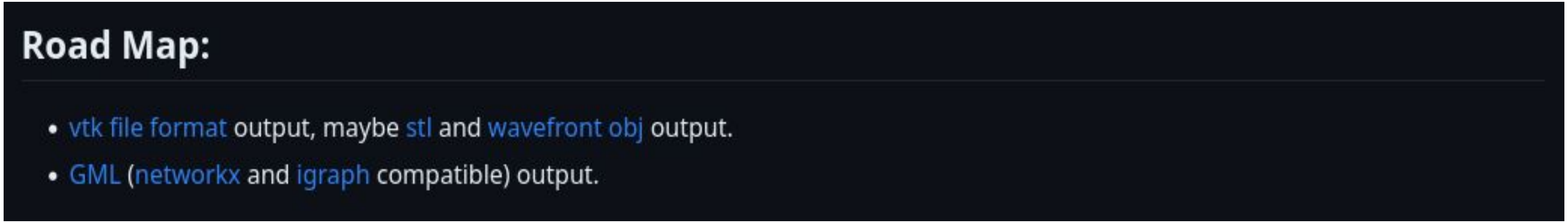
at https://scverse.org :

- read the anndata docs - the basic data object
- read the scanpy docs - single cell analysis
- read the squidpy docs - spatial single cell analysis
- read the scvi-tools docs - single cell machine learning
- …

at https://github.com/PhysiCell-Tools/python-loader :

- work through the pcdl TUTORIAL.md
- read the docstrings from the “workhorse functions” mentioned in the pcdl REFERENCE.md

pcdl - future development



Road Map:

- vtk file format output, maybe stl and wavefront obj output.
- GML (networkx and igraph compatible) output.

Acknowledgement

The whole **MathCancer** lab!

python-loader original implementation:

- Pat Wall
- Randy Heiland
- Paul Macklin

pcdl 2022 / 2023 evolution:

- Marshal Gress
- Heber Rocha

OHSU:

- Jenny Eng (scanpy)
- Tina Ghodsi Asnaashari (abm)

Software - because pcdl is standing on the shoulder of giants:

- PhysiCell,
- PhysiCell Studio
- Python3 core libraries
- numpy, scipy, pandas, matplotlib
- scveres: anndata
- http: requests

Fellowship:

- Chateaubriand - the Office for Science & Technology of the Embassy of France in the United States.

🐧 🐧 🐧

Funding

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)

Python3
PhysiCell
DataLoader