========

Overview

========

Lattice Microbes (LM) is a software package for efficiently sampling trajectories from the chemical and reaction-diffusion master equations (CME/RDME) [#LM1] . LM computes CME/RDME using Graphics Processing Units (GPU) and even multiple GPUs [#LM2] . LM is recently used for the simulation of mRNA splicing in a HeLa cell [#LM3]\_.

LM has been designed for building geometrically-shaped cells. However, LM essentially can incorporate any shapes of cells and cellular structure, as demonstrated in a previous study [#LM4]\_.

LM is designed to simulate cells with geometric shapes, because of limited availability of information on cellular shapes. However, LM originally can incorporate any shapes of cellular boundary or the other cellular structure, as demonstrated in a previous study [#LM4] .

We here developed an extension of LM, which is named Lattice Dendrites (LD), for the efficient incorporation of realistic cellular shapes. In particular, LD is designed to incorporate shapes of segmented images from electron microscopy (EM). LD provides a variety of utility classes/functions for:

#. Conversion of segmented shapes from a software for EM segmentation (reconstruct).

#. Object rotation to minimize their volume in a rectangular solid.

#. Manual annotation using UNI-EM annotator.

#. Surface molecules distribution proportional to surface area.

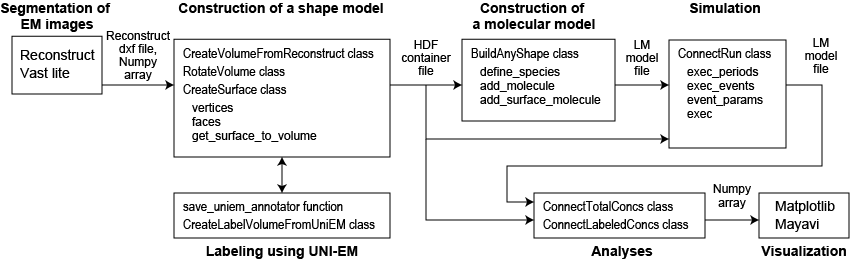
#. Simulation with events

#. Data analyses for visualization.

Users can use a variety of functions below for their model development, simulation, and analyses. Their usages are described in Tutorials 1 and 2. In tutorial 1, we develop a schematic model of dendritic spine, and simulate Ca2+ influx through NMDA receptors and fluorescence recovery after photobleaching (FRAP). In tutorial 2, we introduce a method for the import of realistic shapes of cells from the external software Reconstruct, and annotate it using UNI-EM annotator.

In tutorial 1, we would introduce how to build a geometrically-shaped dendritic spine, and how to simulate Ca2+ influx through NMDA receptors as well as fluorescence recovery after photobleaching (FRAP).

In tutorial 2, we would introduce how to import realistic shapes of cells from the external software Reconstruct, and how to annotate them using UNI-EM annotator.



.. [#LM1] Roberts E, Stone JE, and Luthey-Schulten Z (2013) Lattice Microbes: high-performance stochastic simulation method for the reaction-diffusion master equation, J. Comput. Chem. 34(3):245-255, http://faculty.scs.illinois.edu/schulten/lm/ , http://faculty.scs.illinois.edu/schulten/Software2.0.html#1

.. [#LM2] Hallock MJ, Stone JE, Roberts E, Fry C, Luthey-Schulten Z (2014) Simulation of reaction diffusion processes over biologically-relevant size and time scales using multi-GPU workstations, Parallel Comput. 40:86-99

.. [#LM3] Ghaemi Z, Peterson JR, Gruebele M, and Luthey-Schulten Z (2020) An in-silico human cell model reveals the influence of organization on RNA splicing, PLOS Comput. Biol. 16(3): e1007717, https://eukaryoticcellbuilder.github.io/HeLa\_Builder/

.. [#LM4] Earnest, TM, Watanabe, R, Stone, JE, Mahamid, J, Baumeister, W, Villa, E, & Luthey-Schulten, Z (2017) Challenges of integrating stochastic dynamics and cryo-electron tomograms in whole-cell simulations. J. Phys. Chem. B, 121(15):3871-3881

==============================

Tutorial 1: schematic dendrite

==============================

We would introduce how to use LD and LM through two tutorials. In this tutorial (tutorial 1), we first build a dendritic spine that is represented by a schematic shape (ball and stick), then two types of simulation are conducted: Ca2+ influx through NMDA receptors and fluorescence recovery after photobleaching (FRAP). Through these simulations, users can understand how to build a spiny dendrite, label the spine, run simulation, and visualize the simulation results.

.. toctree::

tut1/create\_spiny\_dendrite

tut1/label\_spine

tut1/set\_reactions

tut1/run\_simulation

tut1/visualization1\_matplotlib

tut1/visualization2\_mayavi

All tutorial programs are located in $LD\_DIRECTORY/tutorial. Find the following script in the subdirectory "1". It provides utility functions that are used in tutorial 1. The function "make\_cylinder" makes a cylinder with an indicated radius and length. The function "add\_shape" locates a geometric object into a volume space. The function "show\_dendrite" loads and displays the surface objects of cellular boundary, postsynaptic density (PSD), mitochondrion, and endoplasmic reticulum (ER).

All tutorial programs are located in $LD\_DIRECTORY/tutorial. Find the following script in the subdirectory "1". It includes utility functions for tutorial 1. The function "make\_cylinder" makes a cylinder with a indicated radius and length. The function "add\_shape" locates such a geometric object in a volume space. The function "show\_dendrite" loads and displays the surface objects of cellular boundary, postsynaptic density (PSD), mitochondrion, and endoplasmic reticulum (ER).

.. literalinclude:: ../tutorial/1/tut1\_functions.py

:language: python

:linenos:

:caption: tut1\_functions.py

==============

Create a shape

==============

1. We would first draw a blueprint of a spiny dendrite as a combination of geometric shapes (Figure above). A sphere with a radius of 0.25 |mgr| m represents a spine head, part of which is labeled as a postsynaptic density (PSD; red colored area). The spine head has a cylindrical spike neck (radius: 0.1 |mgr| m, length 1.0 |mgr| m), which is connected to a parent dendrite that also has a cylindrical shape (radius: 0.5 |mgr| m, length 1.8 |mgr| m). Not only the contour of spiny dendrite, we also introduce two types of intracellular organelles. Endoplasmic reticulum (ER) is set as thin cylinders (radius: 0.08 |mgr| m, length 1.8 |mgr| m; blue colored objects), and a mitochondrion is set as a thich cylinder (radius: 0.2 |mgr| m, length 1.8 |mgr| m; yellow colored objects).
2. The designed shape is embedded in a voxel space for simulation (11\_create\_dend.py). First, a voxel space of 96 |times| 60 |times| 96 voxels (20 nm/voxel) is set as a numpy 3D array 'vol\_dend' (Line 14). Then, the spine head is created using a Python module function morphology.ball (skimage; Line 10; 0.25 |mgr| m, 12 voxels), which is added to 'vol\_dend' (add\_shape; Line 15). Similarly, we make the spine neck and dendrite as cylinders (Lines 11 and 12, respectively), and add them to 'vol\_dend' (Lines 16 and 17, respectively). The in-house function 'add\_shape,' simply overlays overwrapped regions, and represents the filled areas as 1 (cytosolic region) and the void areas as 0 (extracellular region). Thus, we have already built the contour of the spiny dendrite in 'vol\_dend.' Similarly, the PSD, Mitochondrion, and ER are embedded in vol\_psd, vol\_mito, and vol\_er, respectively.
3. Here, the volume sizes need to be corrected. LM can only simulate a volume with a size of a multiple of 32 |times| 32 |times| 32 voxels, whereas the current voxel size is 96 |times| 60 |times| 96 voxels. We thus execute a utility function 'lmpad.' The voxel sizes are automatically expanded to the multiple of 32 (96 |times| 64 |times| 96 voxels in this case). Users can of course set a multiple of 32 voxels from the beginning.
4. Finally, the surface of each object is ER and mitochondrion are also added to

and the designed shape is embedded in a voxel space.