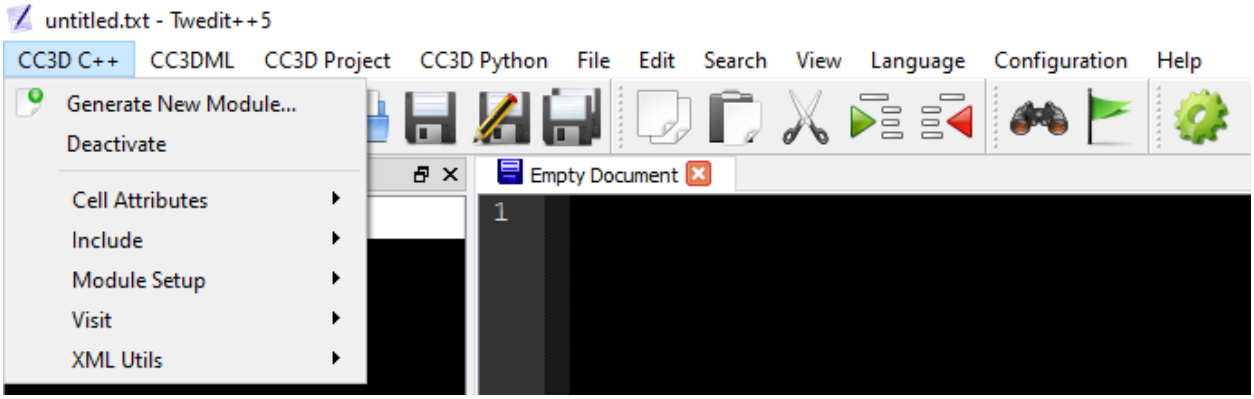
Contents

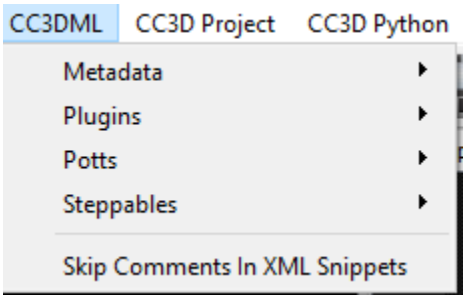
Twedit++ Menu Options

CC3D C++



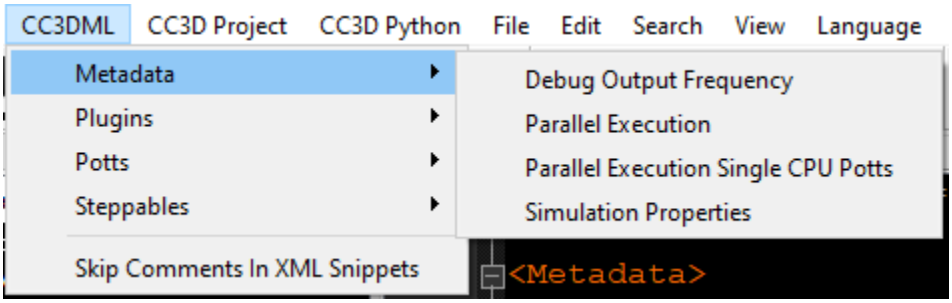
Commands to help with creating and linking to a new module written in C++.

CC3DML



XML snippets for loading / configuring CompuCell3D

Metadata



Debug Output Frequency:

How often energy calculation / spin flip data is printed to console

Parallel Execution:

Use multiple cores/threads to execute

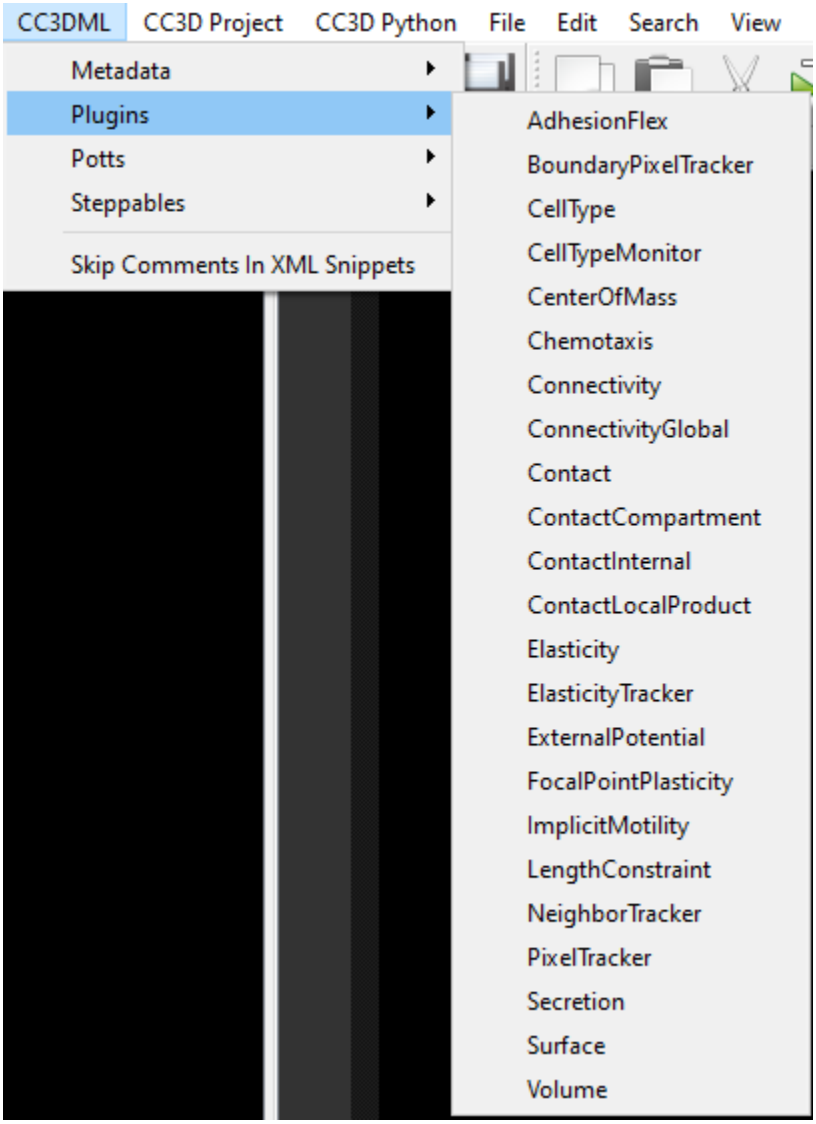
Parallel Execution Single CPU Potts:

Parallelize everything but the Potts algorithm (avoids issues with big cells on parallelized Potts)

Simulation Properties:

Pastes code block that includes everything above

Plugins



XML snippets to load and configure all available plugins

AdhesionFlex

Determines adhesion energy through molecular expression quantities

BoundaryPixelTracker

Enables tracking of cells' boundary (surface) pixels

CellType
Determines cell types

CenterOfMass
Enables cell center of mass tracking

Chemotaxis
Enables chemotactic directional movement

Connectivity / ConnectivityGlobal
Avoids cell fragmentation

Contact / ContactCompartment / ContactInternal / ContactLocalProduct
Determines contact energies between cell types

Elasticity

ElasticityTracker

ExternalPotential
Allows application of forces to cells (*e.g.* gravity)

FocalPontPlasticity
Allows creation of links between cells, applying forces between them. The default force is a spring force.

ImplicitMotility
Makes cells more motile with the default behavior, also has a “persistent motility” mode that makes cell movement more directional.

LengthConstraint
Constrains cell length, changing the cell aspect ratio

NeighborTracker
Allows tracking and iteration through cells’ neighbors

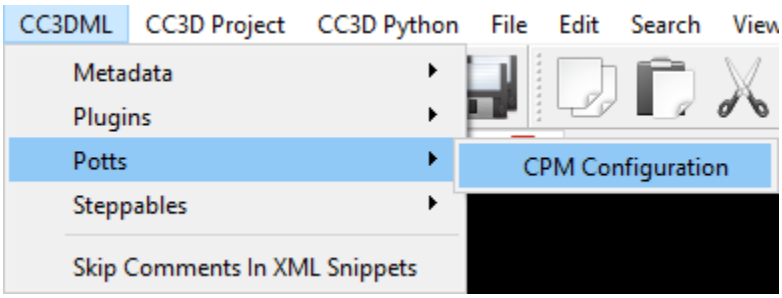
PixelTracker
Enables tracking of cells’ pixels

Secretion
Enables cells secretion / absorption of chemicals

Surface
Constrains cell surface (3D) / perimeter (2D)

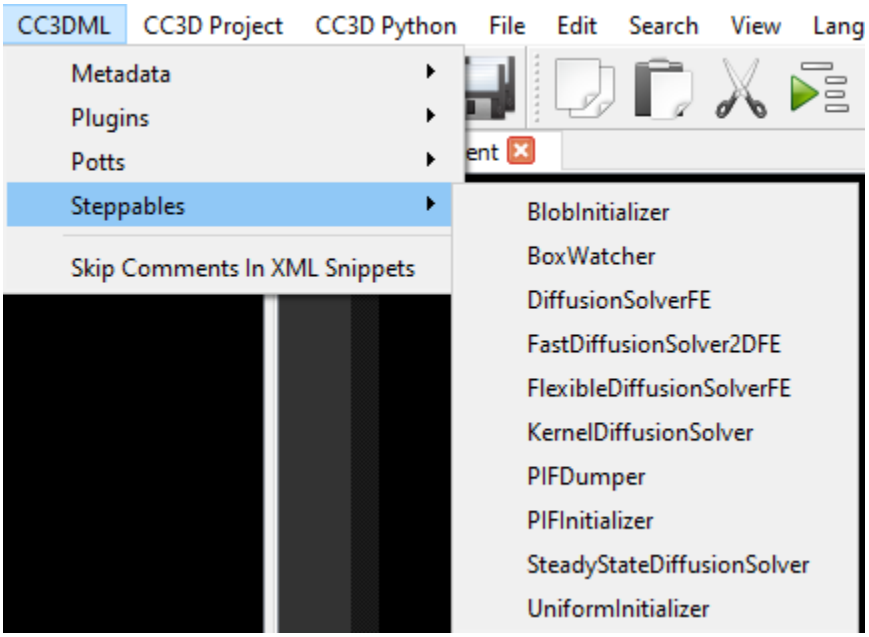
Volume
Constrains cell volume (3D) / area (2D). Without it cells will disappear.

Potts



Configuration of the Cellular Potts algorithm (lattice size, temperature, flip distance)

Steppables



XML snippets to load and configure all available steppables.

BlobInitializer

Initializes cells in a circular zone.

UniformInitializer

Initializes cells in a rectangular zone.

BoxWatcher

Minimizes simulation area under watch to a box around the cells (intended to make simulation faster).

DiffusionSolverFE / FastDiffusionSolver2DFE / KernelDiffusionSolver / SteadyStateDiffusionSolver

Diffusion solvers, DiffusionSolverFE is the default and recommended. SteadyStateDiffusionSolver will iterate diffusion until the steady state is reached each time-step.

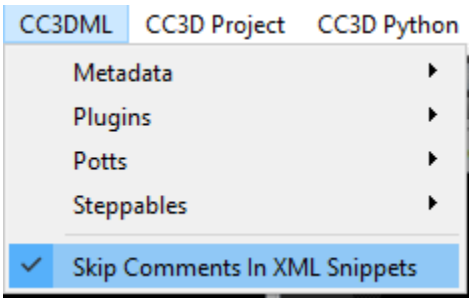
PIFDumper

Creates (dumps) a cell lattice configuration file at specified intervals.

PIFInitializer

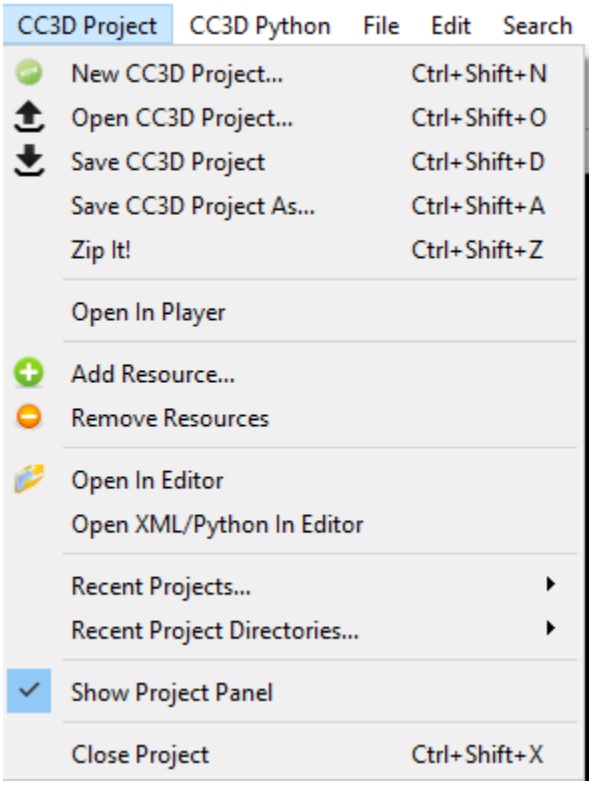
Initializes cell lattice from a PIF file.

Skip Comments in XML Snippets



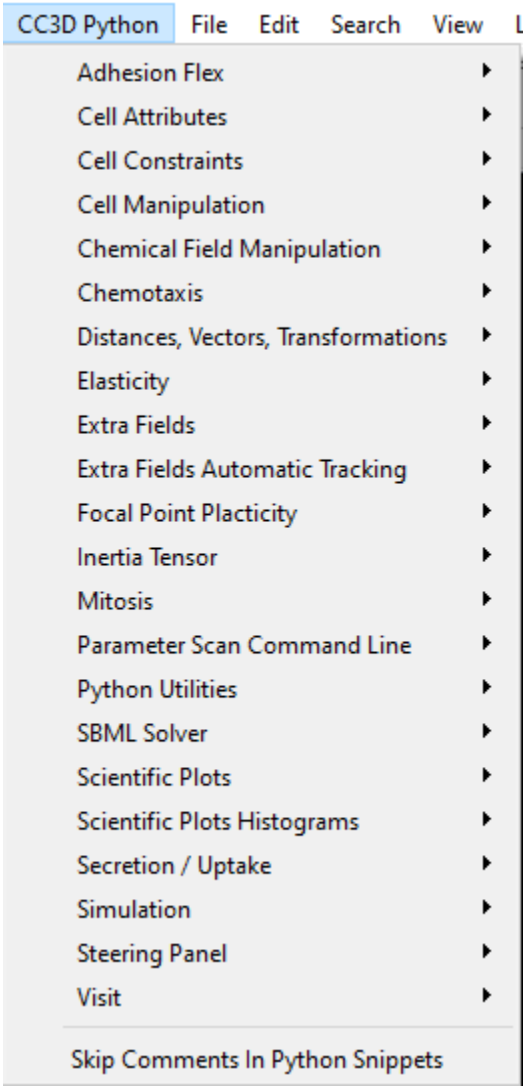
Toggle-able option, skips printing of commented out lines when pasting XML snippets (configuration options that are commented out *will also not* be printed)

CC3D Project



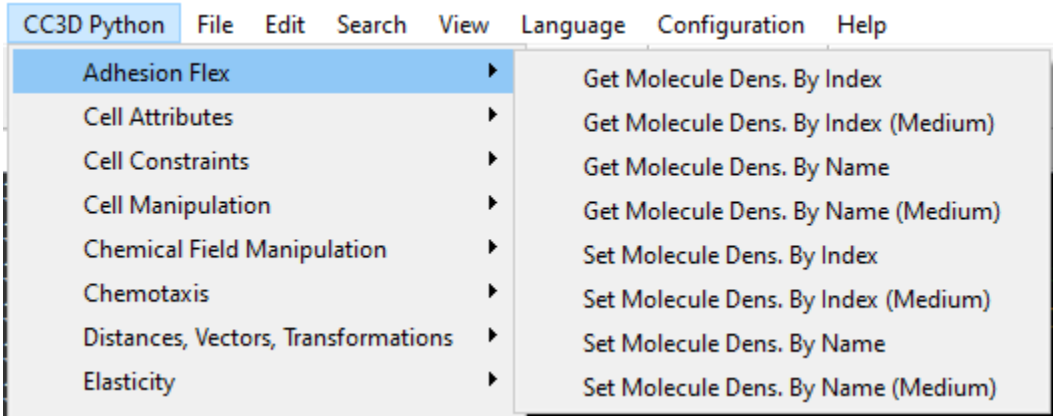
General project file editing of the whole project, *i.e.* the *.cc3d* file and all files used by the simulation. Add/Remove Resource adds or removes files that the project will use (all listed in the *.cc3d*), as python scripts, initial concentration files, cell spatial-configuration files (*.pif*).

CC3D Python



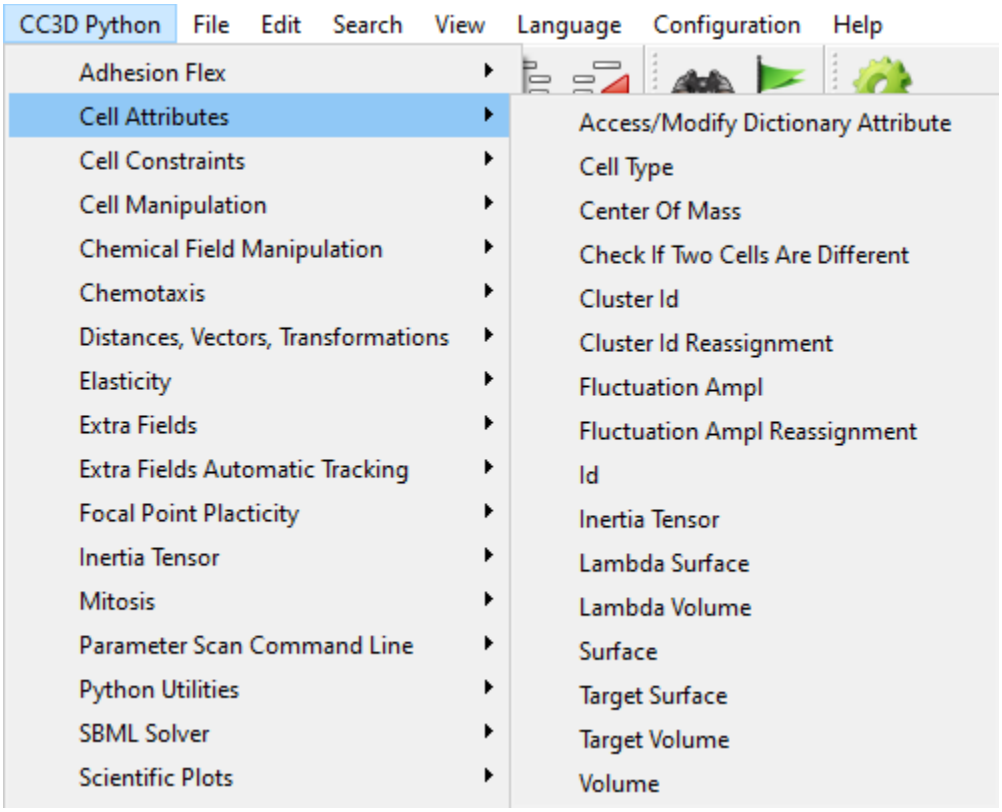
Python snippets for CompuCell3D python functions

Adhesion Flex



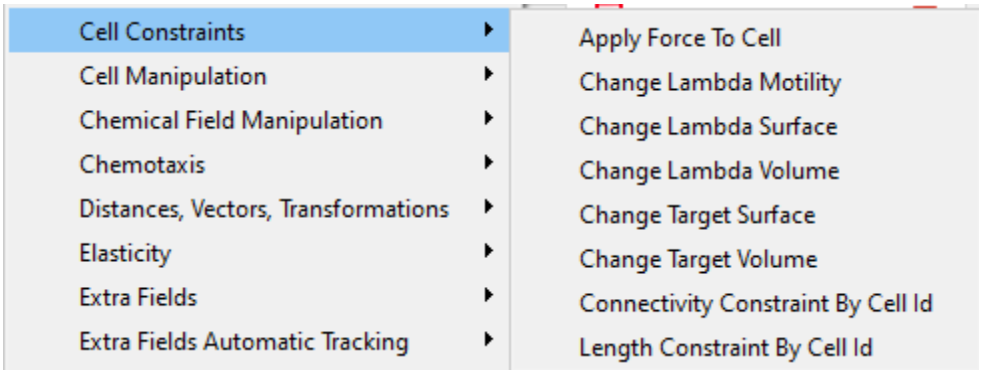
Access/Change attributes of Adhesion Flex Plugin (cell based, should most likely be used in a loop over cells). Contact molecules with medium need to be set in a slightly different manner, they are separated because of it.

Cell Attributes



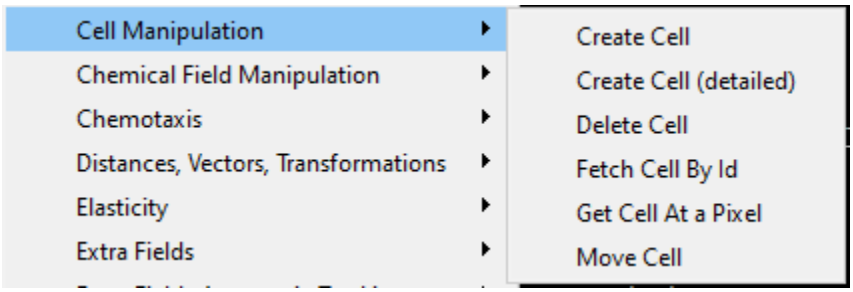
Access / Change cell attributes. Most likely be used in a loop over cells.

Cell Constraints



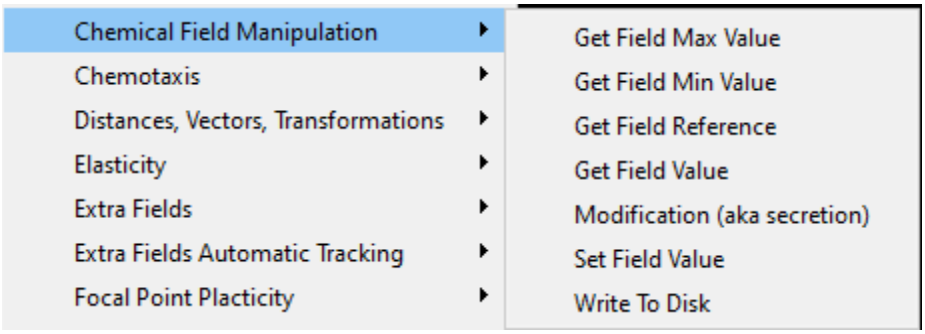
Access / Change energy constraints (plugin parameters) applied to the cell. Most likely be used in a loop over cells.

Cell Manipulation



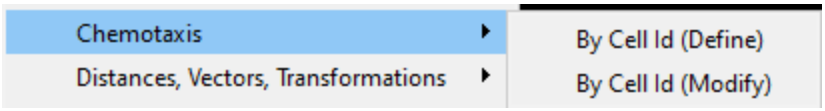
Create/Delete Cells. Access a particular cell (by position or ID). Move whole cell (teleportation).

Chemical Field Manipulation



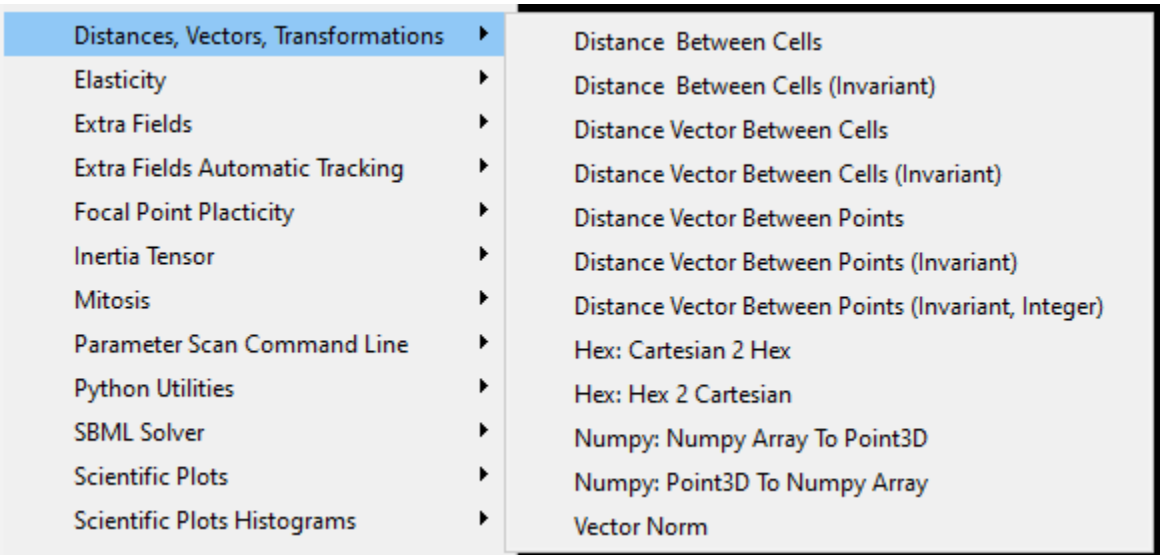
Use the field name for operations. Get field max/min values. Get a reference to the field (create an alias). Get/Set the concentration (field values). Save the field values to a file.

Chemotaxis



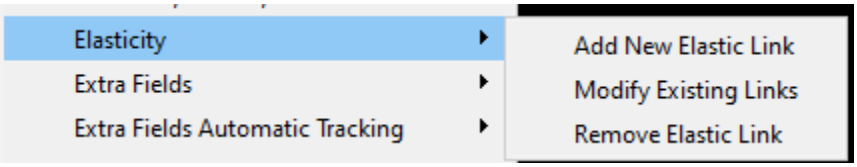
When doing chemotaxis by cell ID first you need to attach chemotactic capabilities to the cell (Define). The parameters can then be changed (Modify).

Distance, Vectors, Transformations



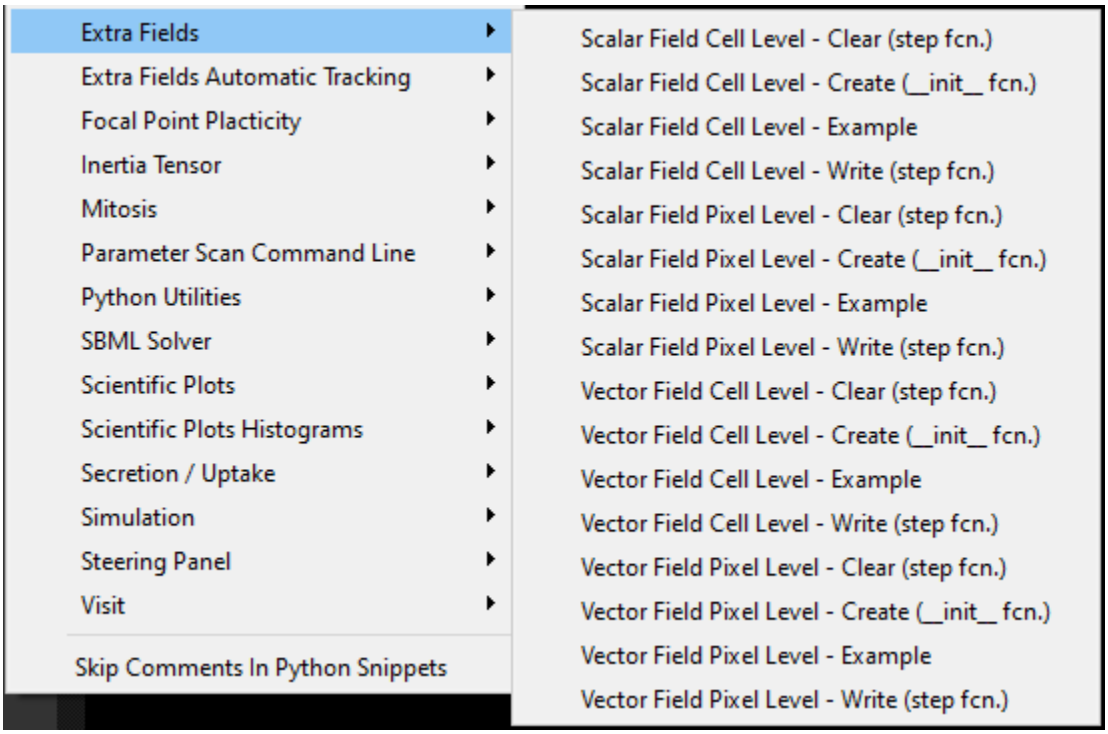
Functions to calculate distances, manipulate coordinates (transforming a square lattice coordinate to a hexagonal lattice one, for instance). Define a vector point in a way CompuCell3D (Point3D) understands versus Numpy array.

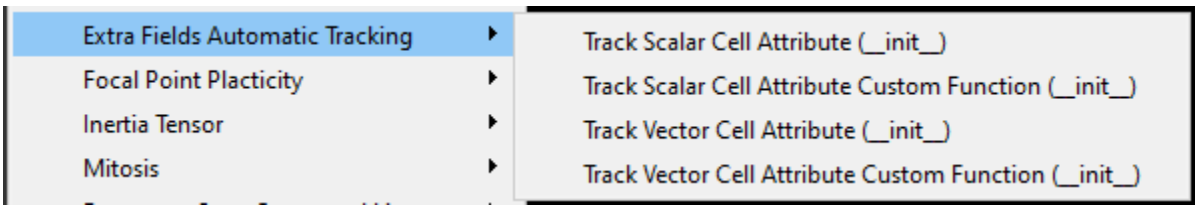
Elasticity



Manipulate elastic links (spring forces) between cells.

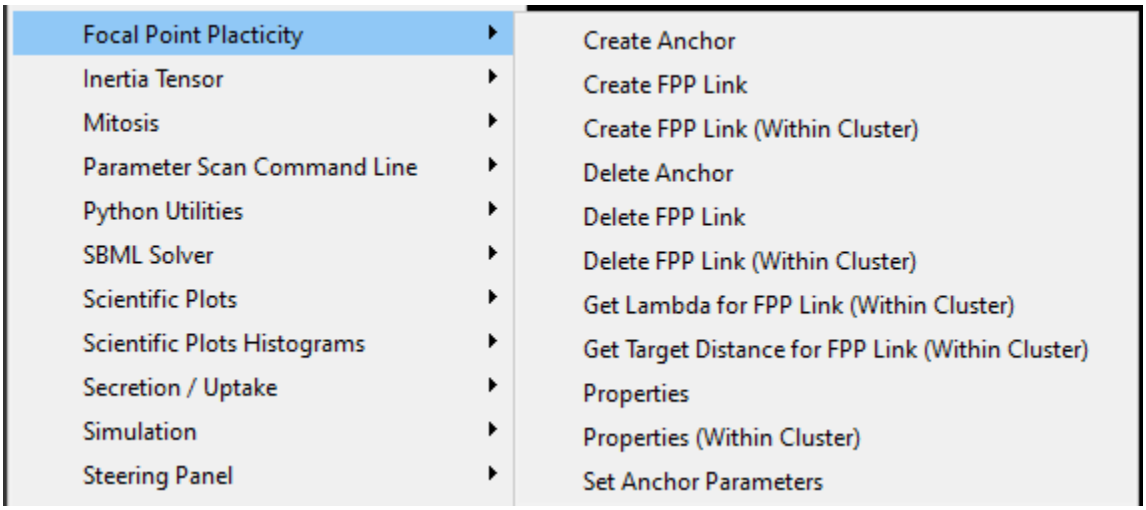
Extra Fields & Extra Fields Automatic Tracking





Create and use “extra fields”. Used to visualize cell attributes (including any added to a cell dictionary). If the attribute is a scalar a heat map will be made from the values and the cells will be colored accordingly. In the case of a vector attribute a vector will be overlaid on the cell and its color will be a heat map of the magnitudes.

Focal Point Plasticity

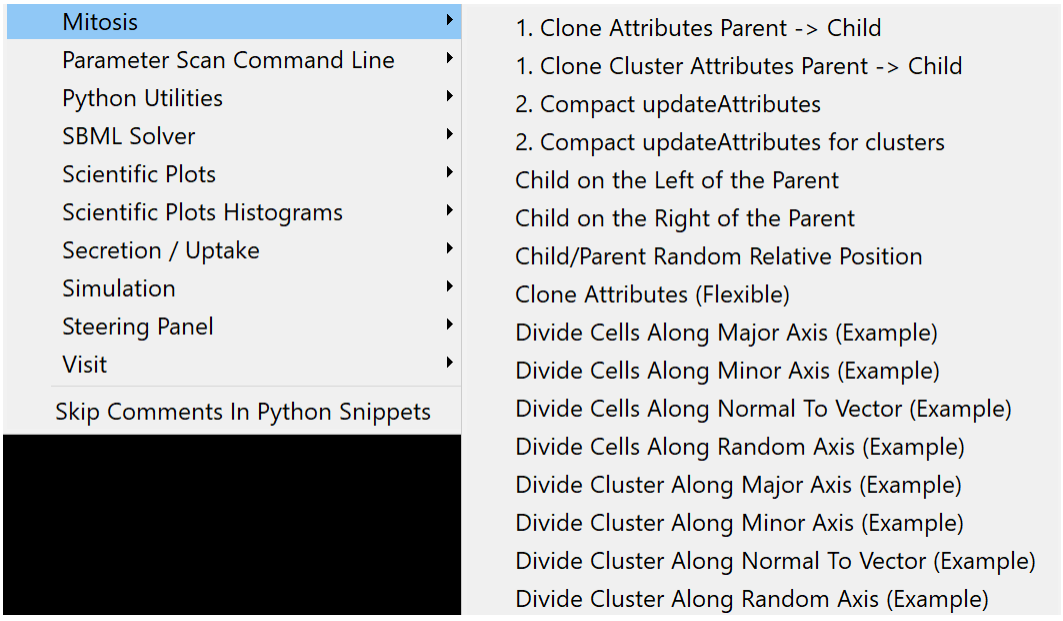


Functions to manipulate Focal Point Plasticity Plugin parameters and of the links themselves.

Inertia Tensor

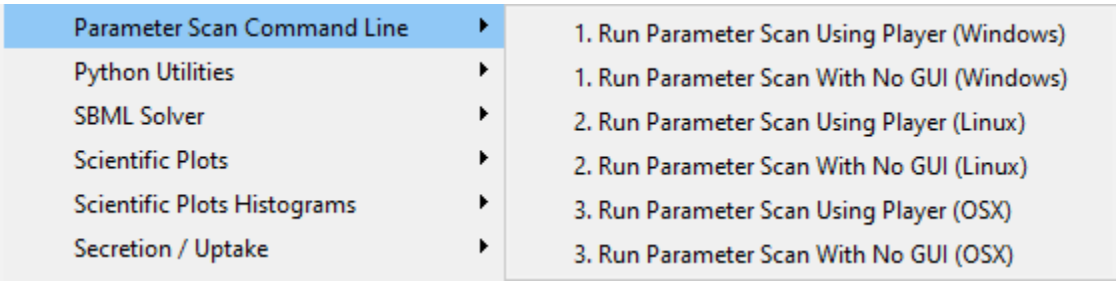
Accessing information from the Tensor of Inertia.

Mitosis



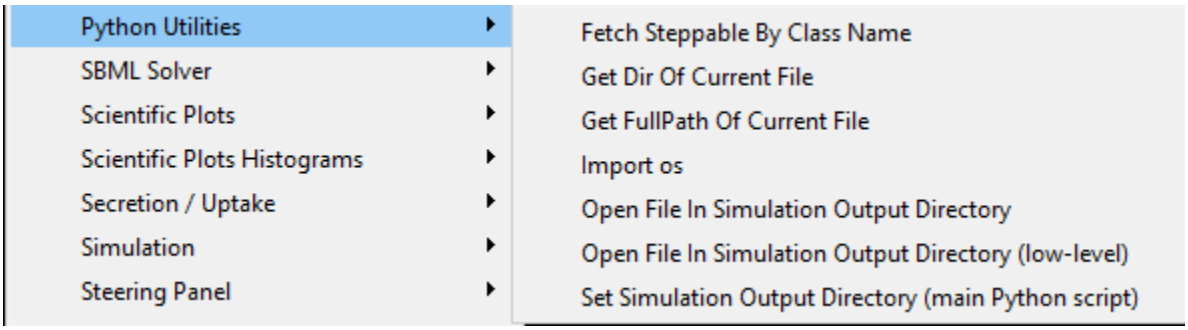
Snippets and examples for using mitosis (cell division) functions. For options with numbers, only one of a particular number should be used, in the numeric order, and at least one of each number is usually required.

Parameter Scan Command Line



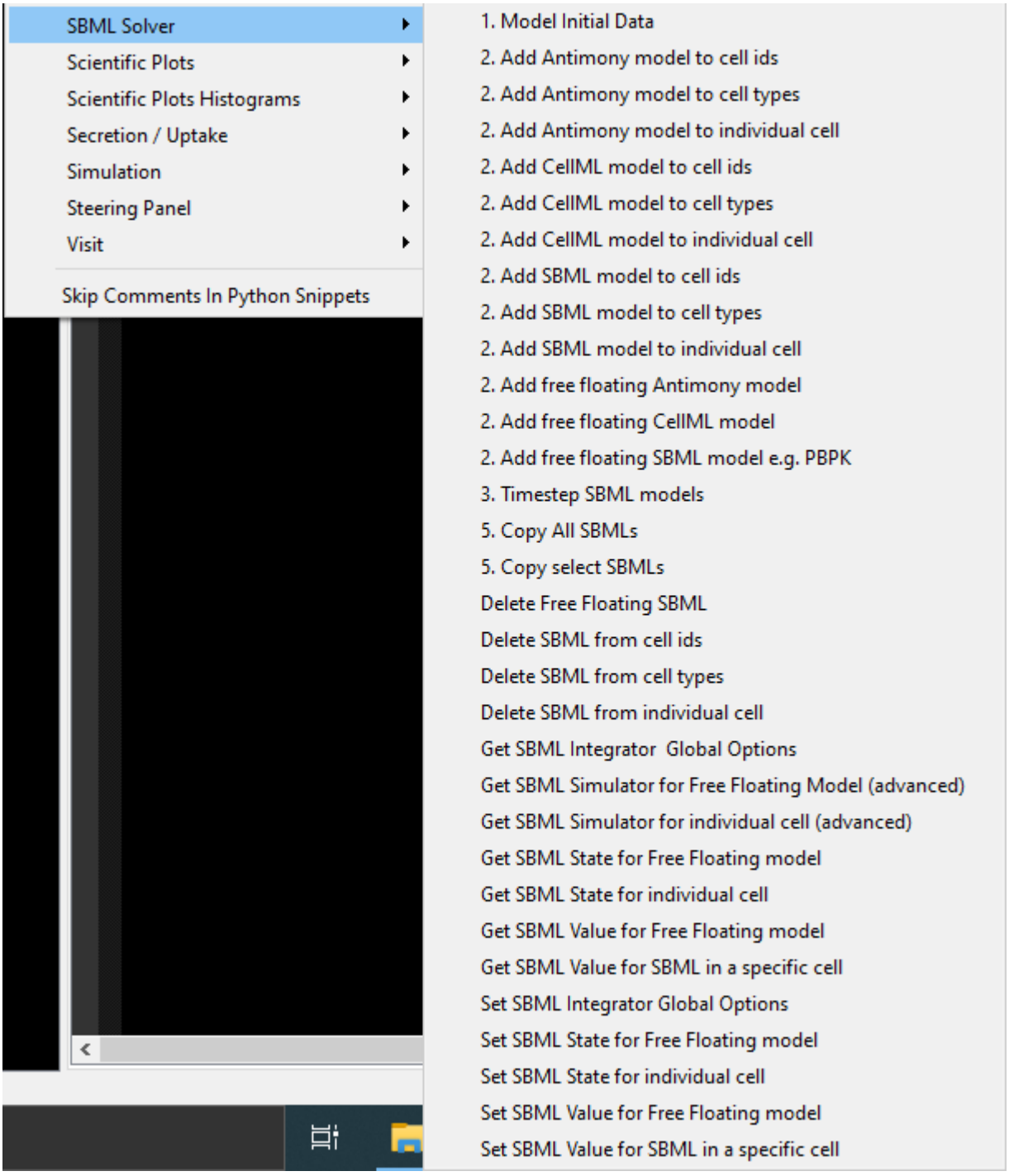
These are intended for use in a .bat (Windows) or .sh (Linux, Mac) file, not in python code. Has the snippet for calling CompuCell3D's parameter scan. For options with numbers, only one of a particular number should be used, in the numeric order, and at least one of each number is usually required.

Python Utilities



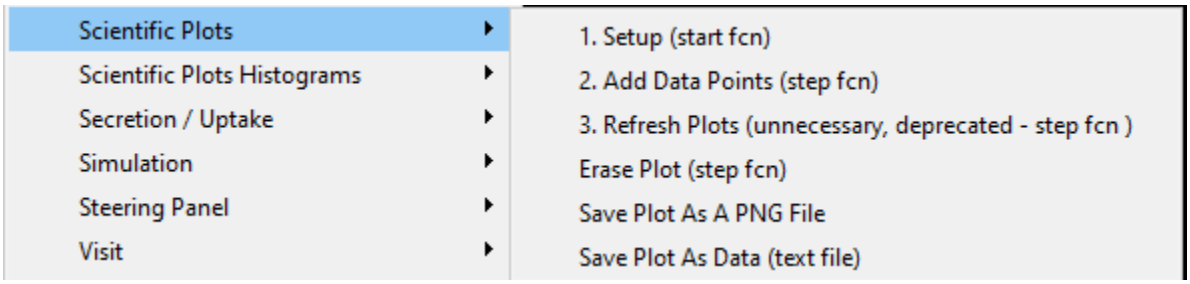
Several python utility snippets, like fetching a pointer to another steppable, getting the directory of the simulation file, setting the output directory.

SBML Solver



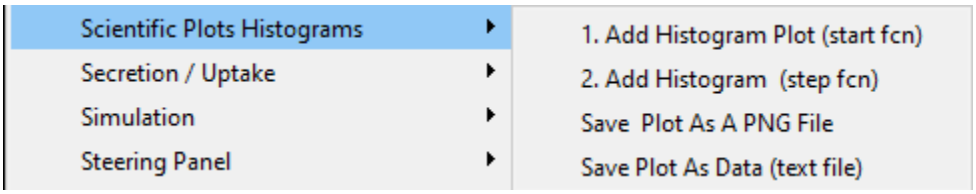
Several functions to set up (load) an SBML model, access it's reactions/variables states, time step it, and so on. For options with numbers, only one of a particular number should be used, in the numeric order, and at least one of each number is usually required.

Scientific Plots



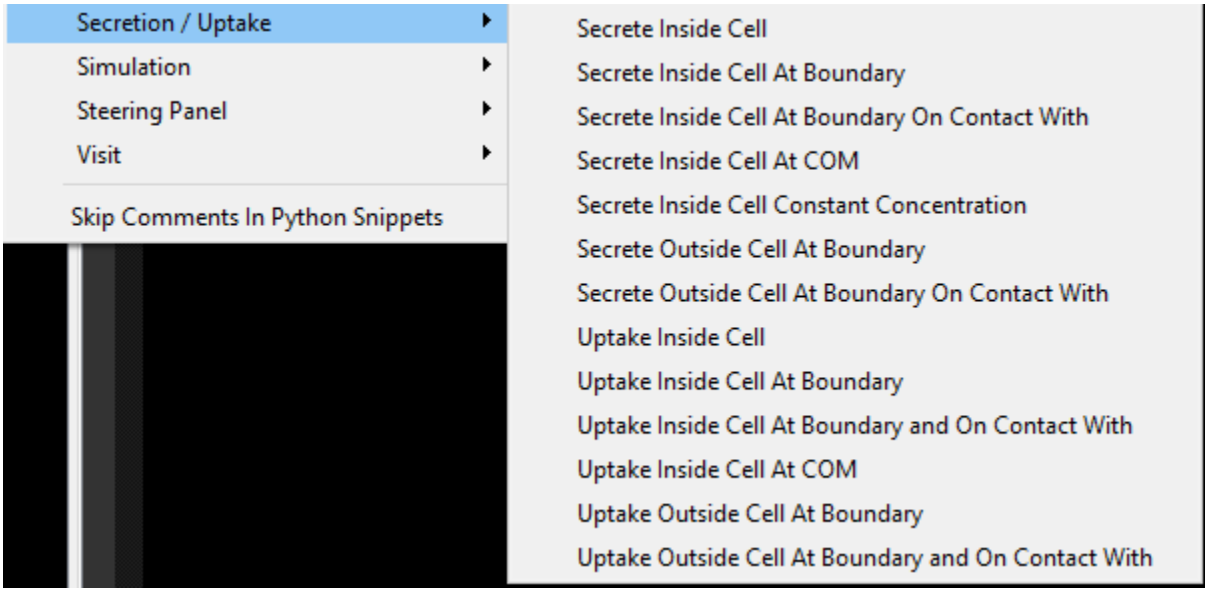
Functions to set up (line/scatter) plots, add data points, save the plots, erase the plotted points. For options with numbers, only one of a particular number should be used, in the numeric order, and at least one of each number is usually required.

Scientific Plots Histograms



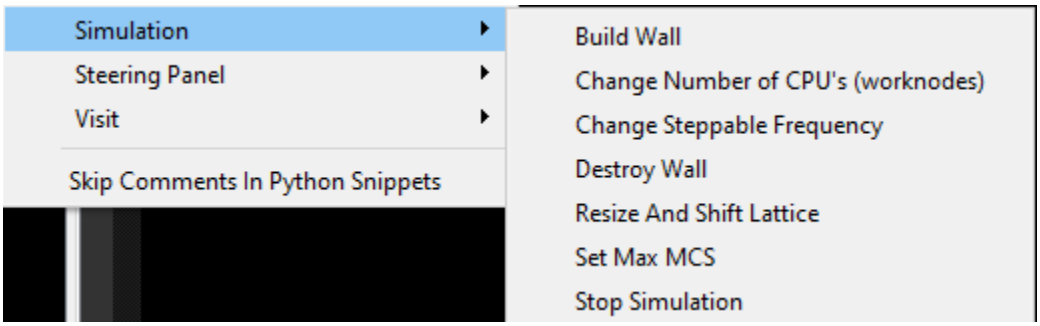
Functions to set up histogram plots, add data points, save the plots. For options with numbers, only one of a particular number should be used, in the numeric order, and at least one of each number is usually required.

Secretion / Uptake



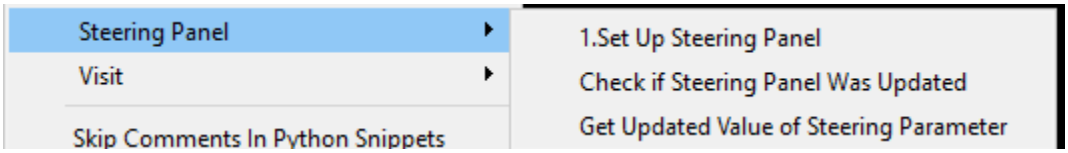
Functions to do secretion / uptake. Uptake is linearly saturated (*i.e.*, grows linearly and then saturates at a constant). Secretion / uptake can be performed over the whole volume of the cell, at its center of mass (COM), at its boundary outside or inside the cell (when selecting “at the boundary” you can also specify that the cell needs to be in contact with another cell of a particular type or with medium).

Simulation



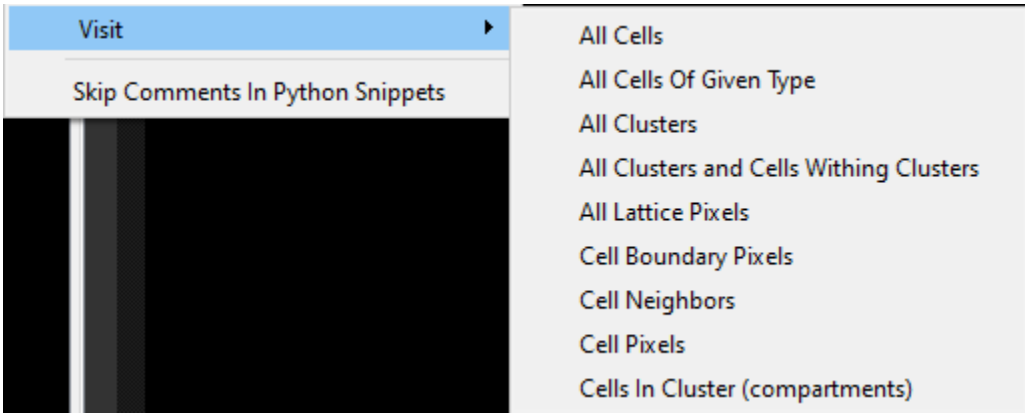
Functions to change how many CPUs or work nodes the simulation should use, how often a steppable should be called, change the lattice size, set how long the simulation should run for (number of time-steps), to stop the simulation early. It also has a function to create (and one to destroy) a border around the simulation lattice (a wall), it is intended to be used with a “frozen” cell. It is useful to do this when using non-periodic boundary conditions so that you can set the contact energy with the border.

Steering panel



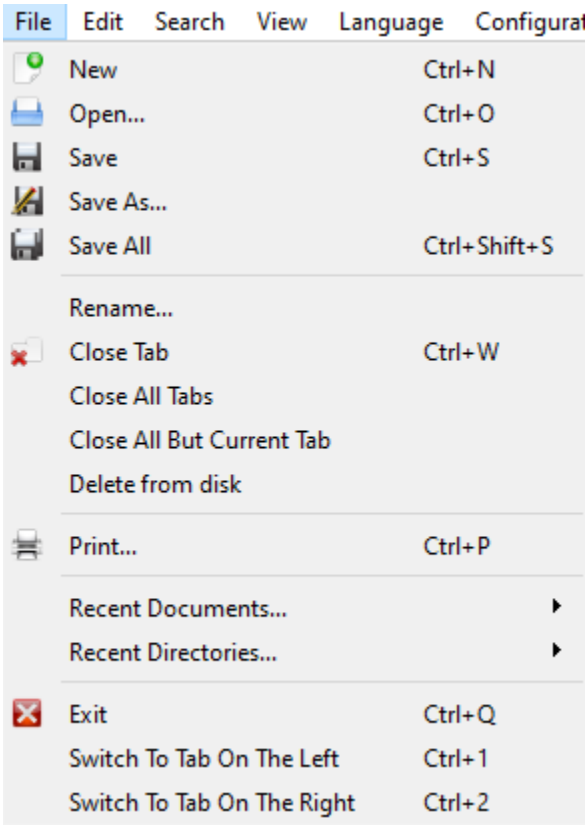
Functions to set up a floating window in CC3D’s player to be used as a steering panel. It can be used to control parameters during run-time graphically. For options with numbers, only one of a particular number should be used, in the numeric order, and at least one of each number is usually required.

Visit

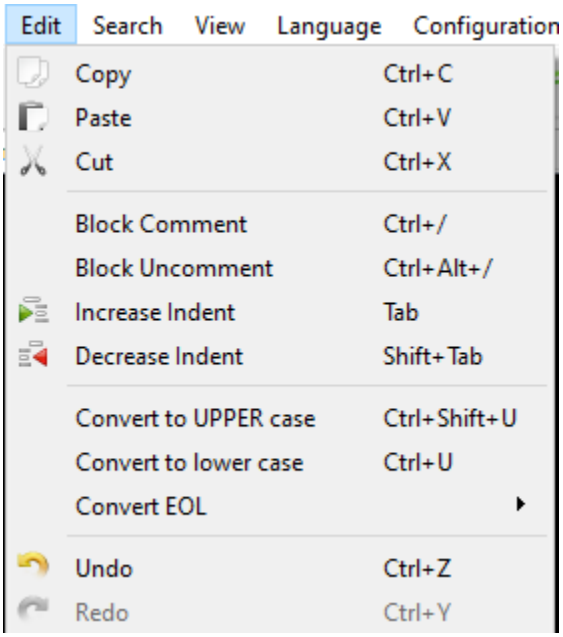


Creates loops over cells, or only a certain type of cells, or cell’s neighbors, or cell clusters etc.

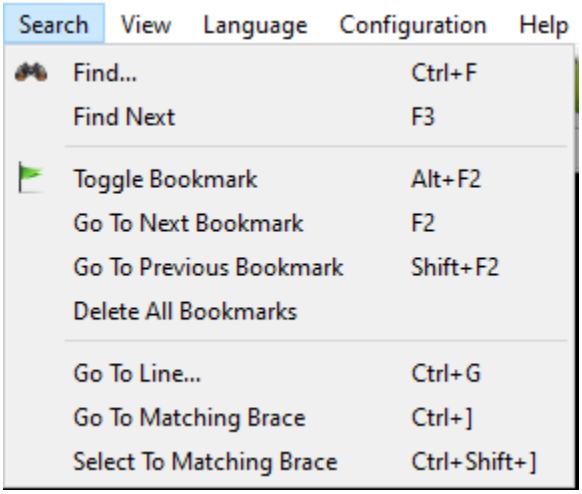
File


File
Edit
Search
View
Language
Configurat
New Ctrl+N
Open... Ctrl+O
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Rename...
Close Tab Ctrl+W
Close All Tabs
Close All But Current Tab
Delete from disk
Print... Ctrl+P
Recent Documents...
Recent Directories...
Exit Ctrl+Q
Switch To Tab On The Left Ctrl+1
Switch To Tab On The Right Ctrl+2

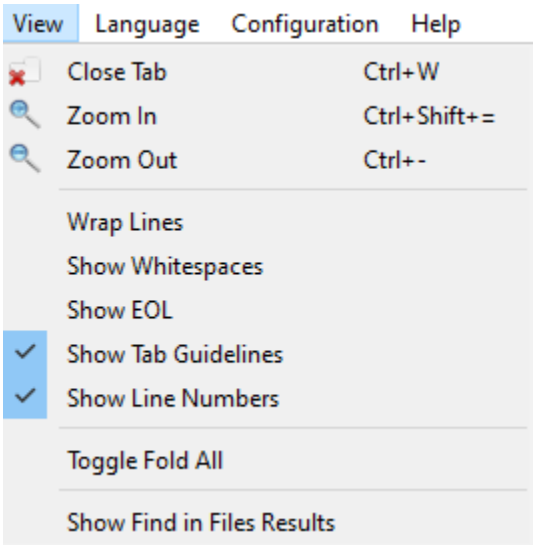
Edit


Edit
Search
View
Language
Configuration
Copy Ctrl+C
Paste Ctrl+V
Cut Ctrl+X
Block Comment Ctrl+/
Block Uncomment Ctrl+Alt+/
Increase Indent Tab
Decrease Indent Shift+Tab
Convert to UPPER case Ctrl+Shift+U
Convert to lower case Ctrl+U
Convert EOL
Undo Ctrl+Z
Redo Ctrl+Y

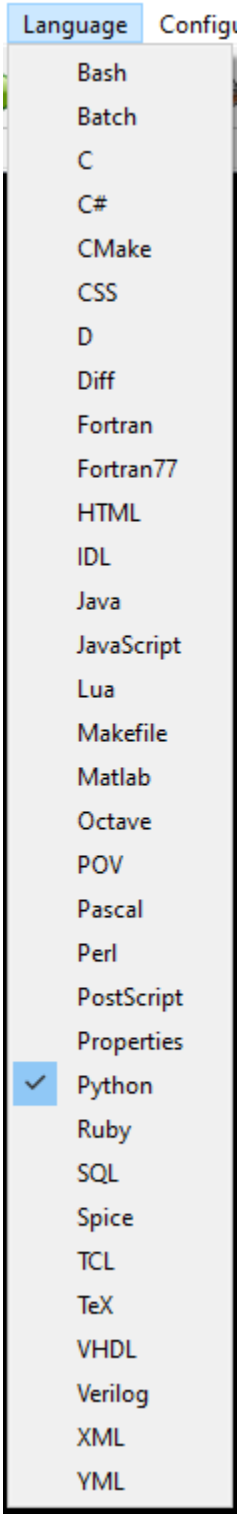
Search


Search
View
Language
Configuration
Help
Find...
Ctrl+F
Find Next
F3
Toggle Bookmark
Alt+F2
Go To Next Bookmark
F2
Go To Previous Bookmark
Shift+F2
Delete All Bookmarks
Go To Line...
Ctrl+G
Go To Matching Brace
Ctrl+]
Select To Matching Brace
Ctrl+Shift+]

View

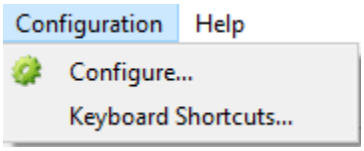

View
Language
Configuration
Help
Close Tab
Ctrl+W
Zoom In
Ctrl+Shift+=
Zoom Out
Ctrl+-
Wrap Lines
Show Whitespaces
Show EOL
Show Tab Guidelines
Show Line Numbers
Toggle Fold All
Show Find in Files Results

Language



Defines the syntax of the currently selected window for syntax highlighting etc.

Configuration



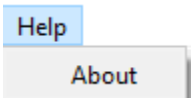
Configure:

Change editing configurations, change the style of Twedit++ (color scheme)

Keyboard Shortcuts:

Change keyboard shortcuts

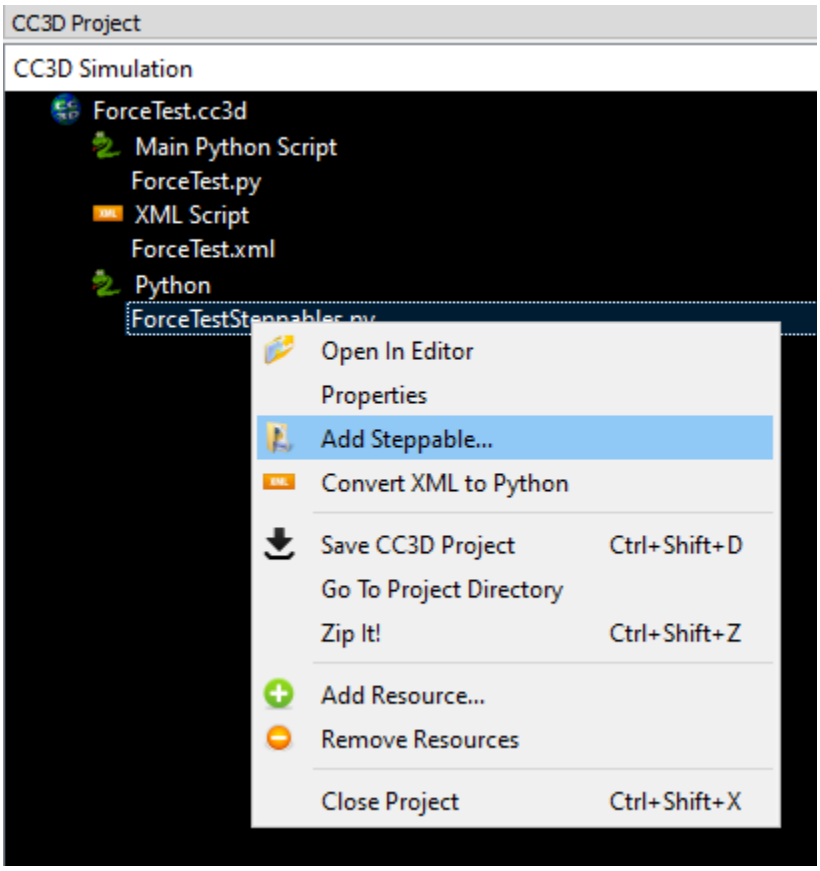
Help



About:

Returns the Twedit++ version number and other information.

Left hand navigation panel right-click



Of note:

Add Steppable:

Opens a window to configure and add a new steppable to the steppables file. Both it and the “main python script” will have code pasted in, so both must be saved.

Convert XML to python:

This converts an xml + python model into an all python version. The xml model specification file is replaced by the equivalent model specification in python.