Blender Visualization Tutorial SS2014 II

CELLmicrocosmos Cell Modeling Project SS2014,
Björn Sommer, Bielefeld University,
Version 03.05.2014

Forum:
http://www.cellvisualization.org

Direct link to this forum entry:
http://www.cellmicrocosmos.org/Cmforum/viewtopic.php?f=21&t=750

Actual Version of Blender:
http://www.blender.org

Here, Blender 2.7a is used.

CELLmicrocosmos 1.1 CellExplorer/4.2 PathwayIntegration
http://cm4.cellmicrocosmos.org

Target

This tutorial describes how to work with the CellExplorer and how to import a cell model generated with the CellExplorer into Blender.

Abbreviation

RMB	Right Mouse Button
LMB	Left Mouse Button

CellExplorer: Open a Cell Model

First make sure that you have installed the latest version of Java on your PC.

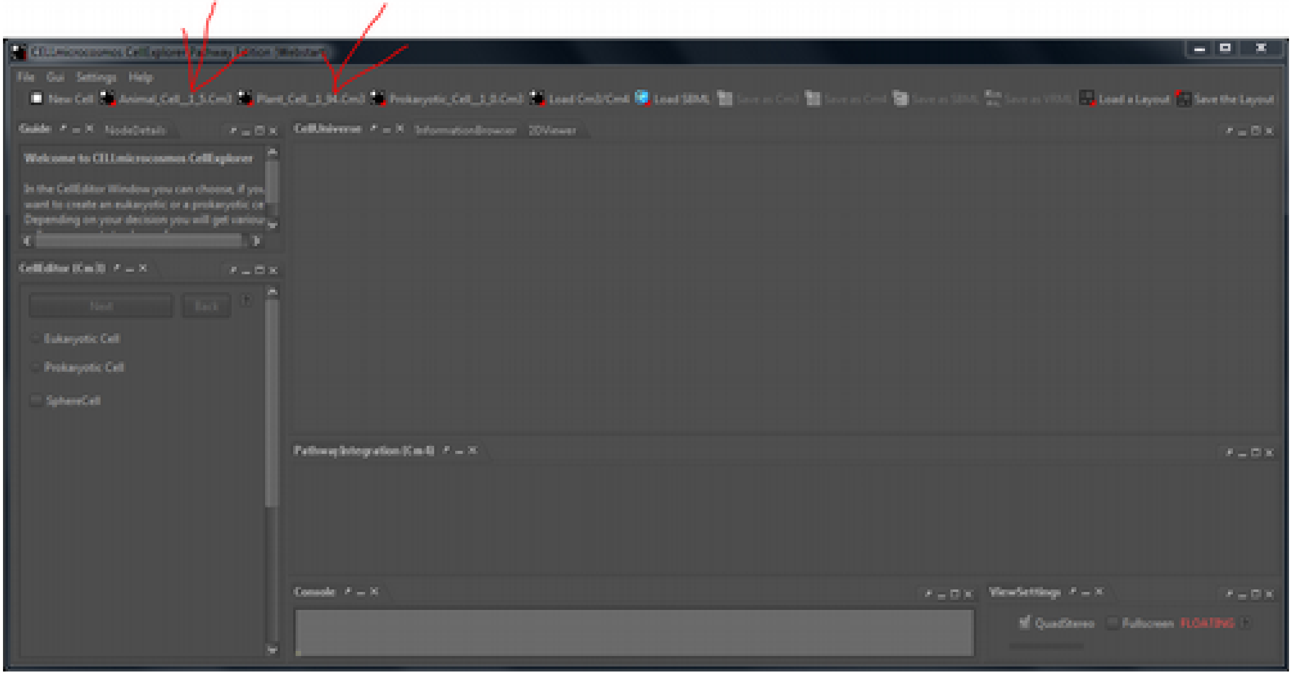
http://java.com

If you are using a 64 bit PC, make sure that you are also installing the 64 bit version of Java providing a much better performance than Java 32 bit! If you are using a Linux system, also the Open Java projects should work with the MembraneEditor, so you do not need the original Sun/Oracle Java.

Then, download and start the CellExplorer (incl. PathwayIntegration) from this website:

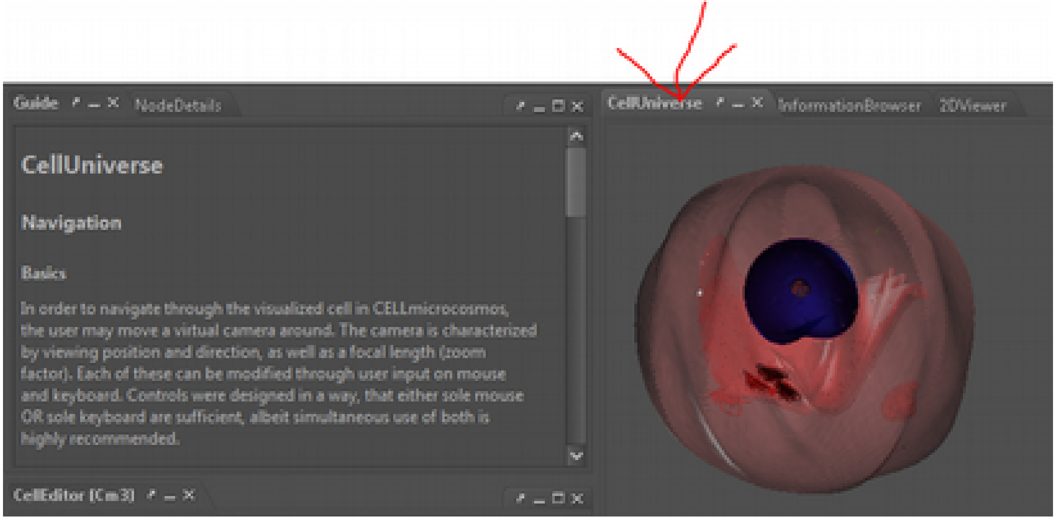
http://Cm4.CELLmicrocosmos.org/

After the program is started you should see something like this:



Click on the first file on the left side and the animal cell is loaded.

To get now information according the navigation etc., just click on the window for which you want to get more information. E.g., click on the Title “CellUniverse” at this



CellUniverse

Navigation

Basics

In order to navigate through the visualized cell in CELLmicrocosmos, the user may move a virtual camera around. The camera is characterized by viewing position and direction, as well as a focal length (zoom factor). Each of these can be modified through user input on mouse and keyboard. Controls were designed in a way, that either sole mouse OR sole keyboard are sufficient, albeit simultaneous use of both is highly recommended.

and have a look to the Guide showing all information about how to navigate through the CellUniverse.

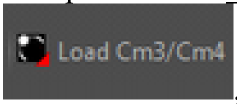
All 3D shapes are based on single VRML97 files. VRML97 was finalized in 1997 and is also known as VRML 2.0 (in contrast to the older version VRML 1.0). It is a text-based format which defines three-dimensional objects. This file format contains color information, vertices and the definition of the edges between these vertices, transparency information, lighting, normals, etc. The VRML97 definition can be found here:

http://www.web3d.org/x3d/specifications/vrml/

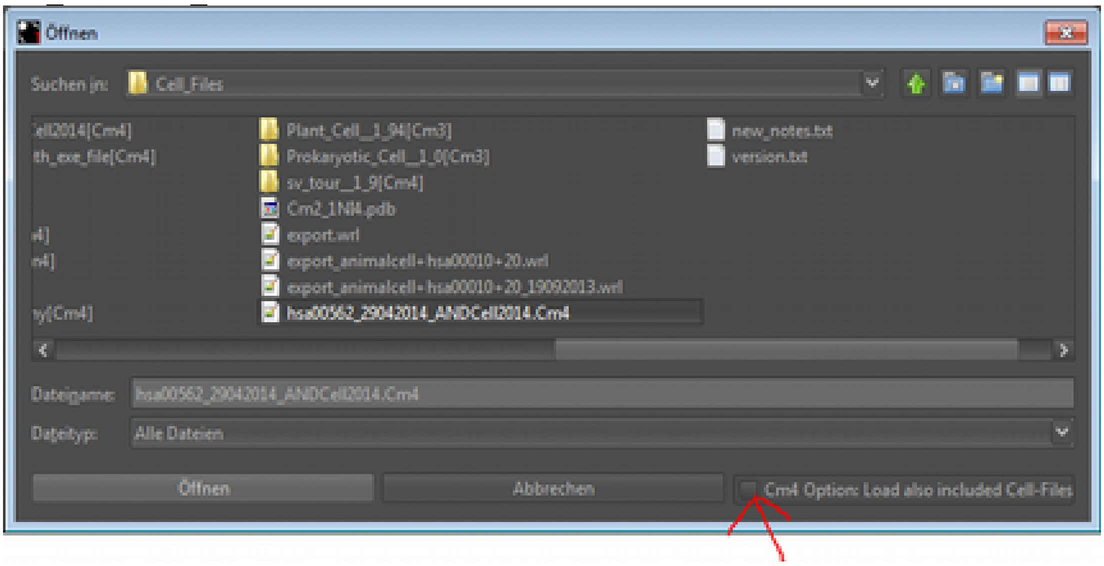
Now, we want to combine this simple cell model with a pathway. First, you have to download the following files from the forum:

http://www.cellmicrocosmos.org/Cmforum/viewtopic.php?f=21&t=737

- Blender_Tutorial_SS2014__2_0files.zip
  - It contains the folder:
    - *io_scene_x3d/*
      - For VRML2 import: Place this folder into the following Blender directory (and substitute the original folder with the same name):
        - *C:\Program Files\Blender Foundation\Blender\2.67\scripts*
  - And the file:
    - *vrml_export_266.py*
      - For VRML2 export.
  - and the Cm4 file: *hsa00562_29042014_ANDCell2014.Cm4*

Unzip “hsa00562_29042014_ANDCell2014.Cm4” and click in the CellExplorer on the button 

, deselect the “Cm4 option”, as shown on the next image, and open the file “hsa00562_29042014_ANDCell2014.Cm4”:



Suchen in: Cell_Files

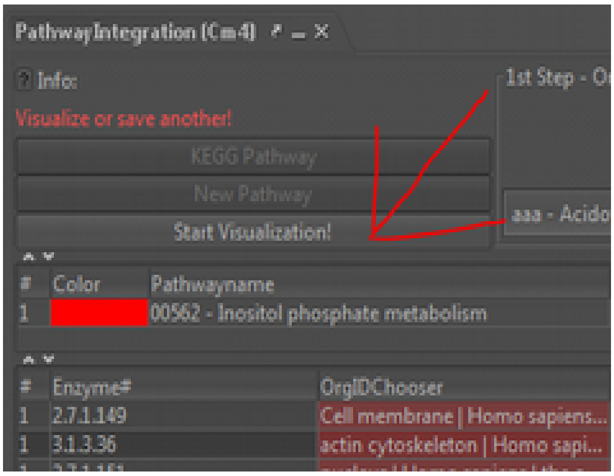
ell2014[Cm4]
th_exe_file[Cm4]
4]
n4]
ly[Cm4]
Plant_Cell_1_94[Cm3]
Prokaryotic_Cell_1_0[Cm3]
sv_tour_1_9[Cm4]
Cm2_1NI4.pdb
export.wrl
export_animalcell+hsa00010+20.wrl
export_animalcell+hsa00010+20_19092013.wrl
hsa00562_29042014_ANDCell2014.Cm4
new_notes.txt
version.txt

Dateiname: hsa00562_29042014_ANDCell2014.Cm4

Dateityp: Alle Dateien

☐ Cm4 Option: Load also included Cell-Files

and then open the file “hsa00562_29042014_ANDCell2014.Cm4”. The file is loaded and now just click on the “Start Visualization” button:

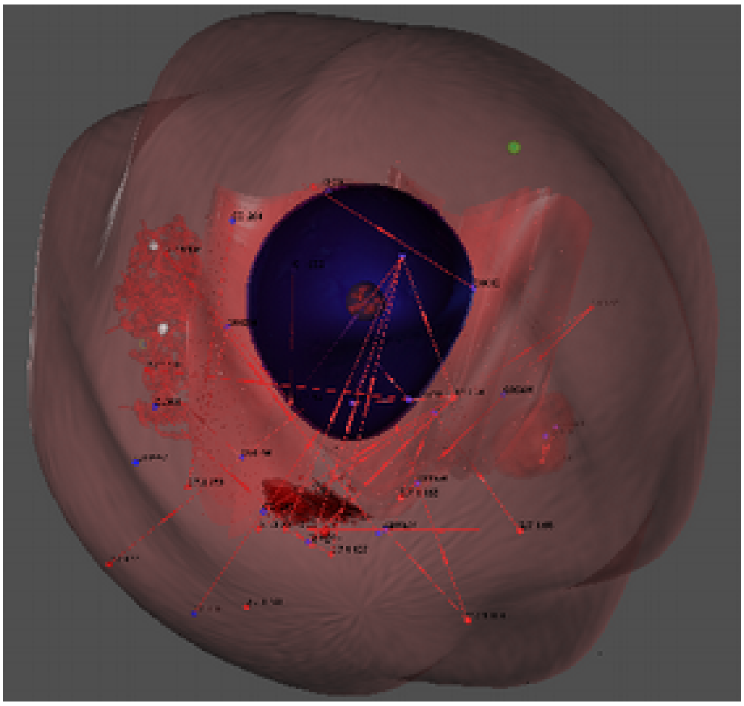


Visualize or save another!

#	Color	Pathwayname
1	🟥	00562 - Inositol phosphate metabolism

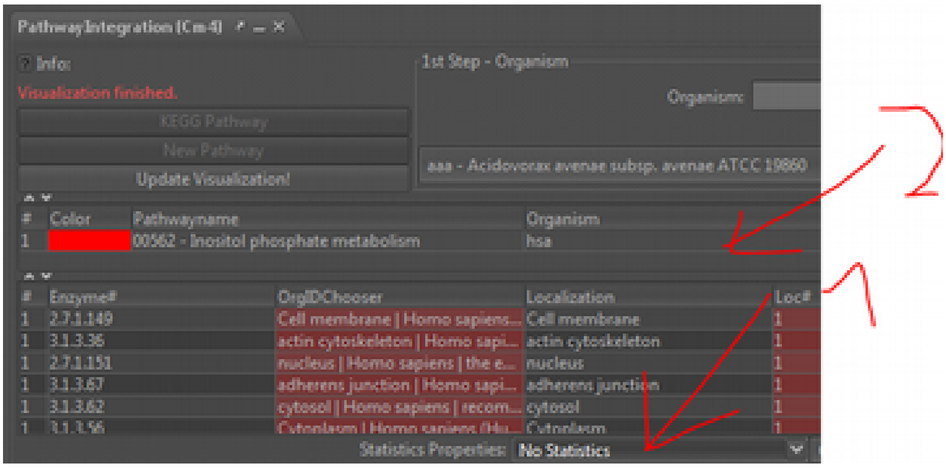
#	Enzyme#	OrgIDChooser
1	2.7.1.149	Cell membrane | Homo sapiens...
1	3.1.3.36	actin cytoskeleton | Homo sapi...

The cell model is now combined with pathway.



The pathway hsa00562 is the “Inositol phosphate meabolism”. It was previously downloaded from our KEGG database integration in the DAWIS-M.D. data warehouse system and then localized by using a set of databases. “hsa” is the abbreviations for homo sapiens, and 00562 is the KEGG-internal number of the pathwa^y.

At the moment, all protein localizations are assigned to different localizations the way they were directly downloaded from the database. To change this, click 1. onto the option button labeled with “No Statistics” and activate the statistics by activating “Localizations”, and then 2. double click onto the actual pathway 00562.



Visualization finished.

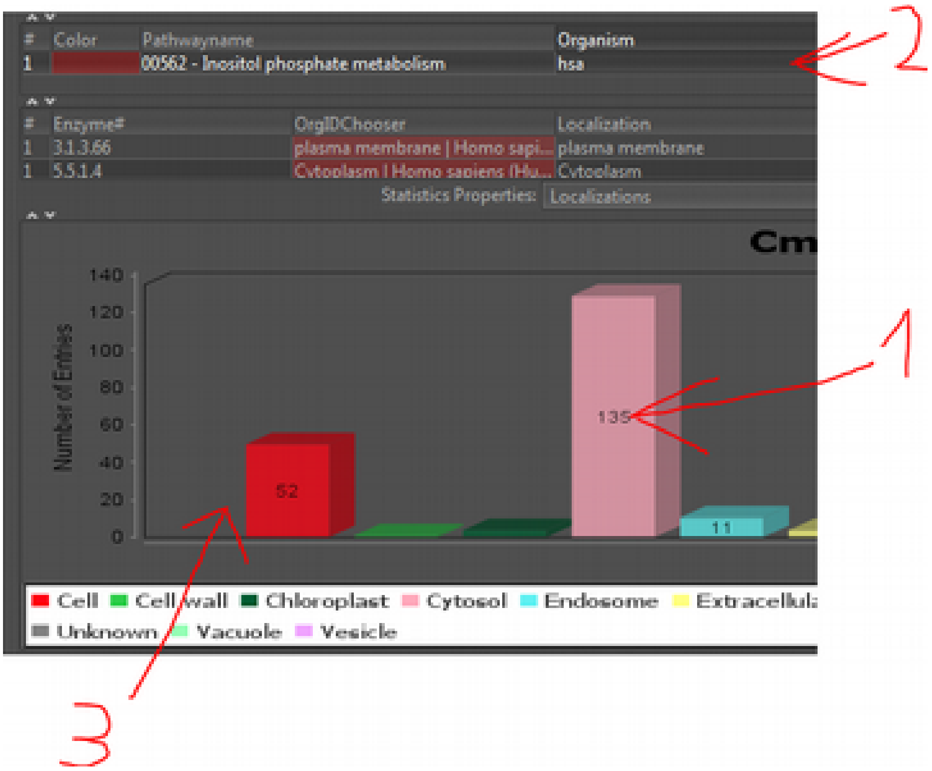
aaa - Acidovorax avenae subsp. avenae ATCC 19860

#	Color	Pathwayname	Organism
1	🟥	00562 - Inositol phosphate metabolism	hsa

#	Enzyme#	OrgIDChooser	Localization	Loc#
1	2.7.1.149	Cell membrane | Homo sapiens...	Cell membrane	1
1	3.1.3.36	actin cytoskeleton | Homo sapi...	actin cytoskeleton	1
1	2.7.1.151	nucleus | Homo sapiens | the e...	nucleus	1
1	3.1.3.67	adherens junction | Homo sapi...	adherens junction	1
1	3.1.3.62	cytosol | Homo sapiens | recom...	cytosol	1
1	3.1.3.56	Cytoplasm | Homo sapiens (Hu...	Cytoplasm	1

Statistics Properties: No Statistics

Then, 1. double-click on the pink chart representing the cytosol, then 2. double-click again onto the pathway 00562 and 3. onto the red chart representing the cell.



#	Color	Pathwayname	Organism
1	🟥	00562 - Inositol phosphate metabolism	hsa

#	Enzyme#	OrgIDChooser	Localization
1	3.1.3.66	plasma membrane | Homo sapi...	plasma membrane
1	5.5.1.4	Cytoplasm | Homo sapiens (Hu...	Cytoplasm

Statistics Properties: Localizations

What did you do? First you have assigned all proteins providing localization information pointing to the cytosol to the cell component cytosol (2^{nd} priority) and then you have assigned all proteins providing localization information pointing towards the cell membrane to the cell component with the same name (1^{st} priority).

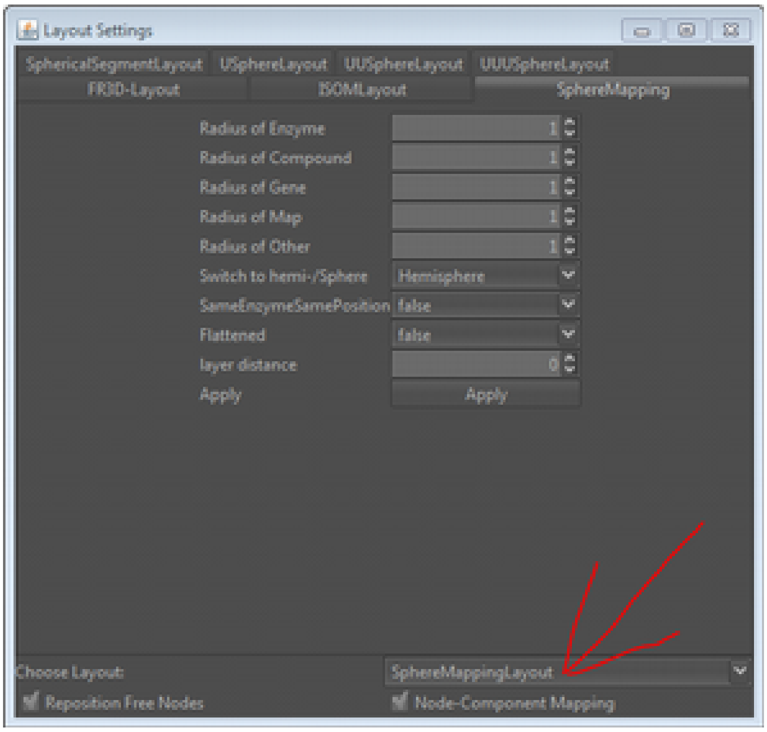
Now, to actualize the 3D visualization, you only have to click again onto “Update Visualization”. The 3D visualization shows now potential protein localizations based on database-acquired data.

One more hint: if only the layout of a 3D pathway should be analyzed, it might make sense to assign all protein localizations to a single cell component. You can use the previously described procedure for this purpose, but you need to activate the “unknown” check box. Otherwise, those localizations which are not based on a database entry will not be activated:

Statistics Properties: Localizations | Cm4 Localizations | ☑ ANDCell ☑ Brenda ☑ GO ☑ Reactome ☑ UniProt ☑ unknown ☐ Selections Only

In this way, it is for example to force the program to map all proteins onto a single specific cell component.

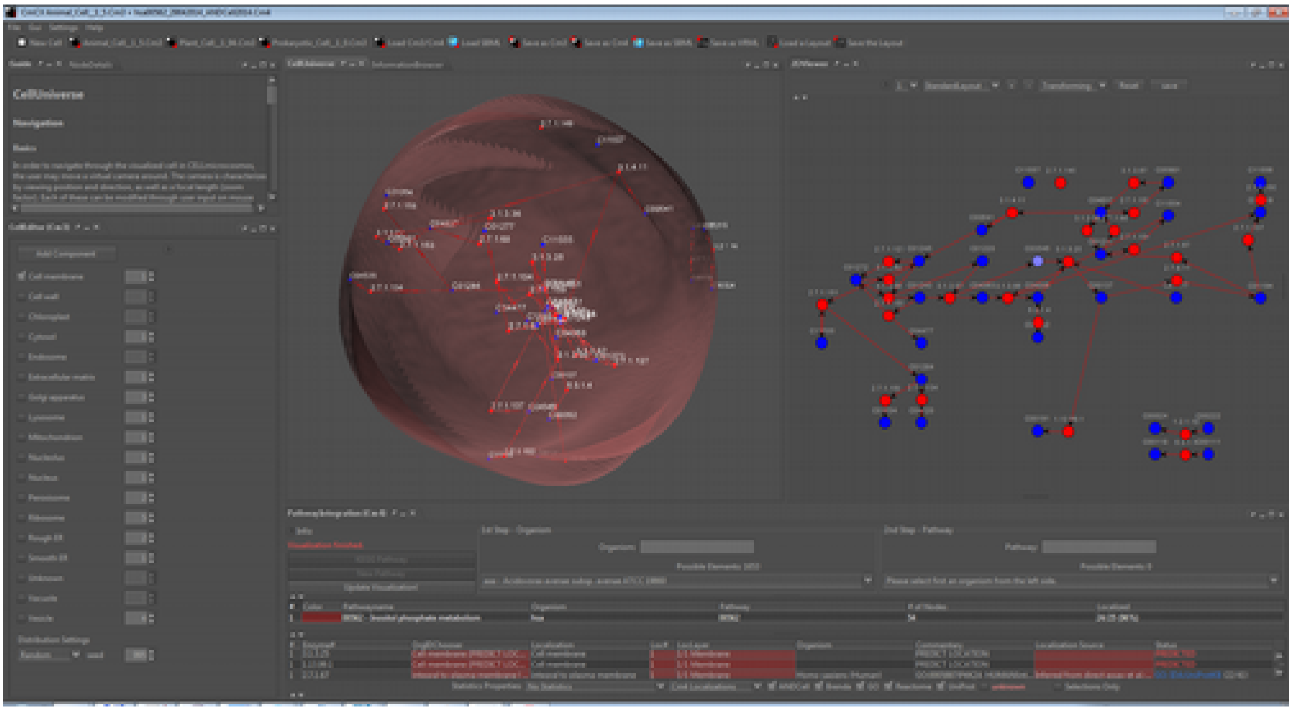
Now, let us choose another node placing algorithm. Open the corresponding dialog by selecting in the menu “Settings” → “Edit Algorithm Settings”. And then, select the algorithm “SphereMappingLayout” and close the window again:



Radius of Enzyme: 1
Radius of Compound: 1
Radius of Gene: 1
Radius of Map: 1
Radius of Other: 1
Switch to hemi-/Sphere: Hemisphere
SameEnzymeSamePosition: false
Flattened: false
layer distance: 0
Apply: Apply

Choose Layout: SphereMappingLayout
☑ Reposition Free Nodes
☑ Node-Component Mapping

Click again “Update Visualization” and you will see that the layout changes.

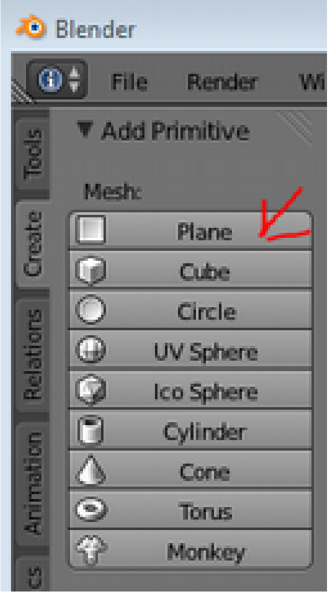


If you want to compare the original layout with the 3D visualization, click and drag the 2DViewer to the right side of the window and then activate the CellUniverse again. You will see now that the SphereMappingLayout places the nodes onto the surface of the cell component by utilizing the 2D coordinates. By the way, in this case, all protein localizations were previously exclusively assigned to the cell membrane.

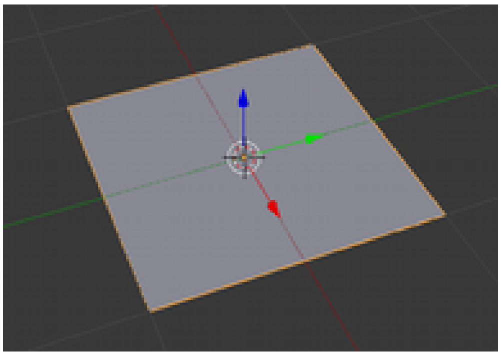
Blender: From Plane to Cell Component

First, delete all 3D objects from the scene with select and DEL (*not* the camera and the light!)

Then, create a plane:



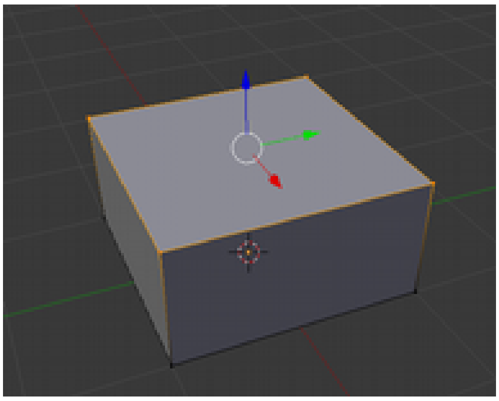
It should be placed in the center of the 3D view



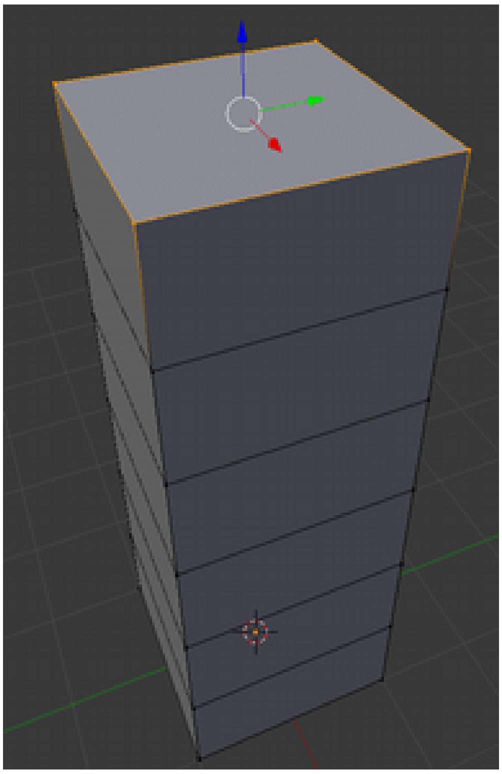
If it is not, you may use the Transform Window (it appears when pressing *N*) to change the Location to 0,0,0.

Now, make sure the plane is selected and change to Edit mode by pressing *TAB*.

The, start to extrude the plane to a 3D object by pressing *E* and moving the mouse pointer upward while pressing *CTRL*. While doing so, move the newly generated side to the first possible position and click LM.

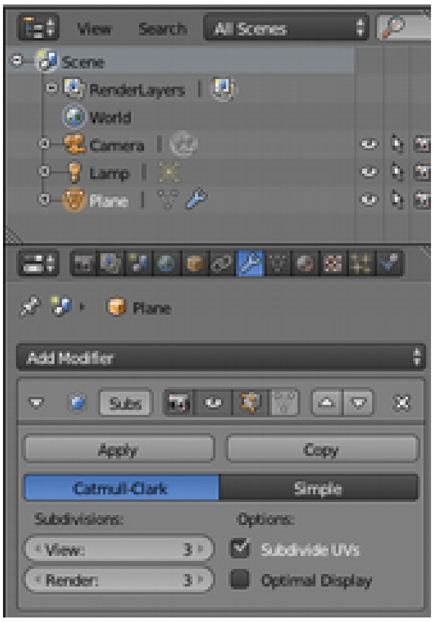


Now, just repeat this process 5 times:



Okay, this does not really look like a cell component, but we will change this now. Leave the Edit Mode by pressing *TAB* again, and now, we will add a modifier.

Make sure, the object is still selected, go to the modifier section in the properties section and add the modifier “Subdivision Surface” (sub section “Generate”):



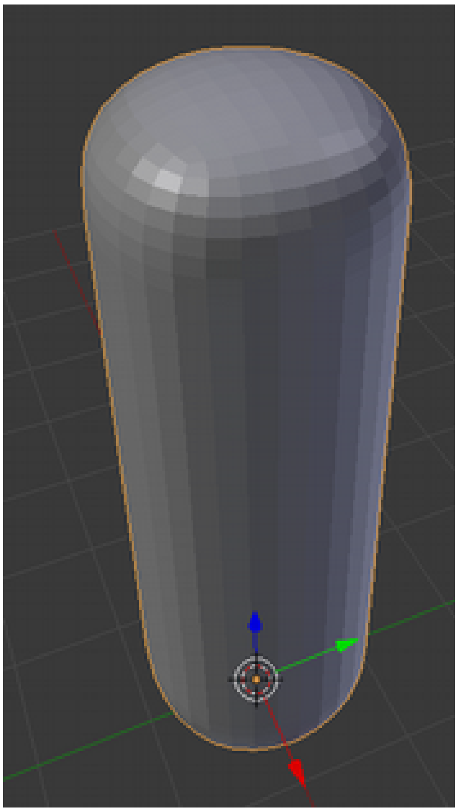
View: 3

Render: 3

☑ Subdivide UVs

☐ Optimal Display

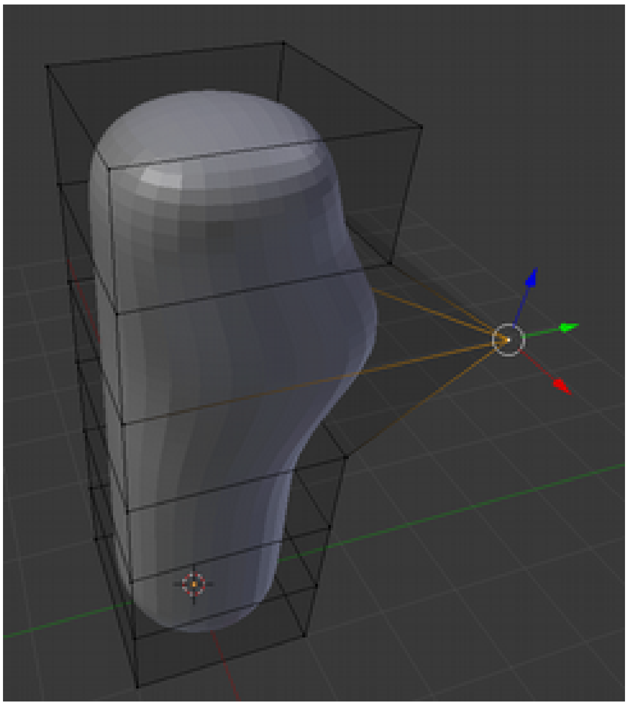
Then, change the “Subdivisions” in view to value 3, and this happens now:



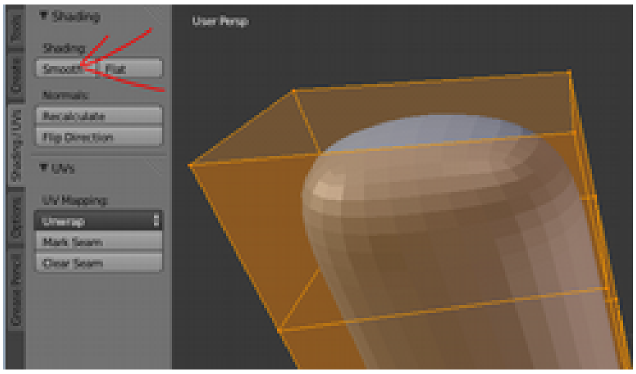
A good starting point fo a mitochondrion, or what do you think?

This is our initial 3D shape. Copy it by using *CTRL+C* and *CTRL+V* two times. The hide two planes:

Now select the actually visual object again and change to the Edit Mode. You will see something like a catch surrounding the tube-like structure. If you select now one point and move it, you will see that the shape smoothly changes.



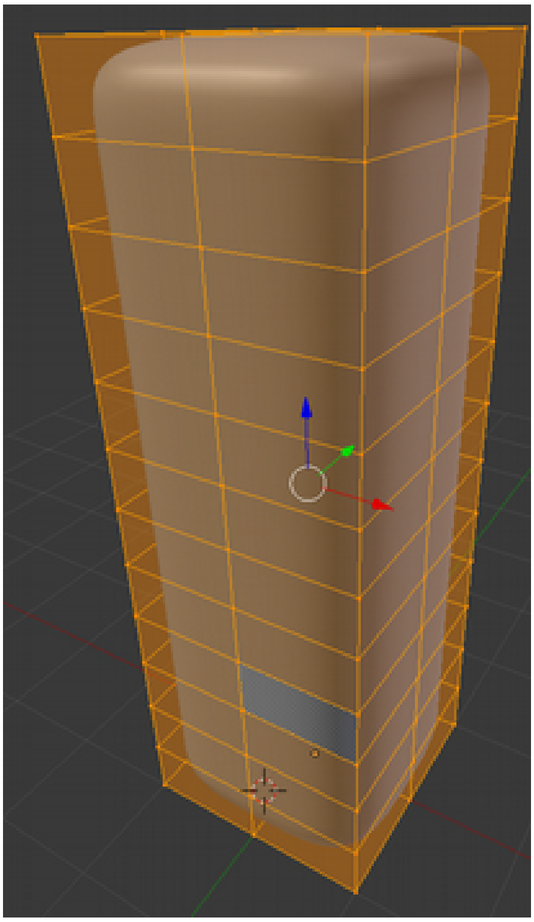
However, you can undo this first (e.g. by pressing *CTRL+Z*). But you still see the grid-like structure on the surface of the tube. Select the whole model by pressing *A* and then change the shading like this (on the Shading/Uvs tab):



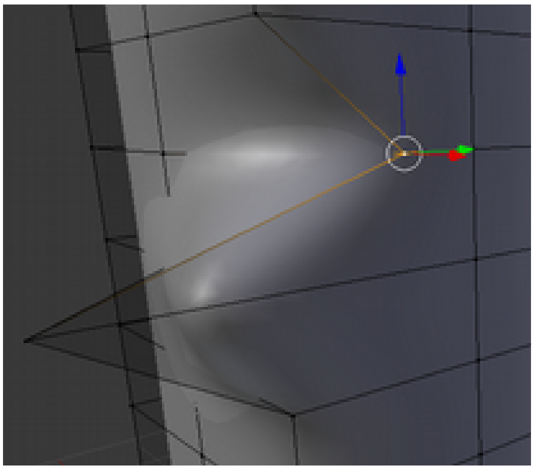
Looks quite good now! For the future, if the surface has a strange look with strange shading, you can try to press here “Normales → Recalculate”.

However, you are now only able to change the outer points of the tube. If you want to make changes more in detail, just do the following:

Select all by pressing *A*, then press *W* and select “Subdivide”.



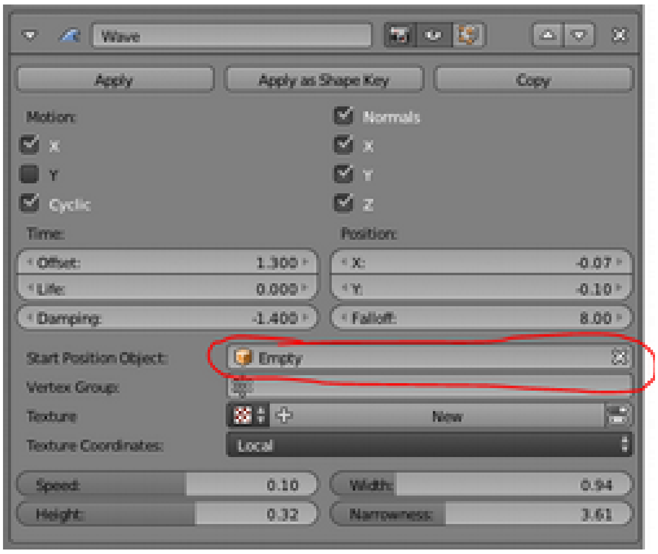
Now you are able to make quite granular changes like this:



But before we start this manual modeling process, let us look at some other modifiers.

We have used the Subdivision Surface modifier. Now, we want to use two other modifier classes, to create dynamically a wave-like, irregular surface structure.

Select the object in object mode and add the modifier “Wave”. Then, choose these settings:



Wave

Motion:
☑ X
☐ Y
☑ Cyclic

☑ Normals
☑ X
☑ Y
☑ Z

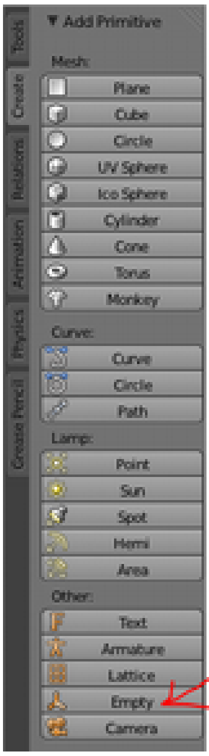
Time:
Offset: 1.300
Life: 0.000
Damping: -1.400

Position:
X: -0.07
Y: -0.10
Falloff: 8.00

Start Position Object: Empty
Vertex Group:
Texture: New
Texture Coordinates: Local

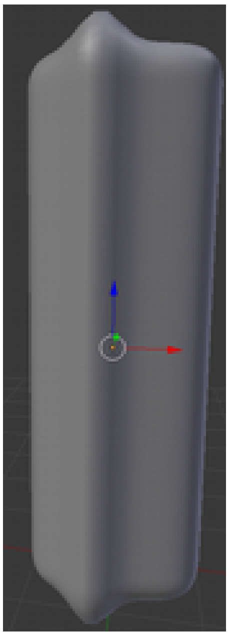
Speed: 0.10
Height: 0.32
Width: 0.94
Narrowness: 3.61

All settings can be changed by using this window, except the “Start Position Object”. Before we can change this, we have to create the empty object by using:



The “Empty” Object is used to define the center of the Wave effect. This object is abstract, it will not be rendered. After generating this object, you have to choose it in the Wave modifier window as the “Start Position Object”. Select the newly created Empty object and move it to the center of your cell component object:

A wave-like structure appears.

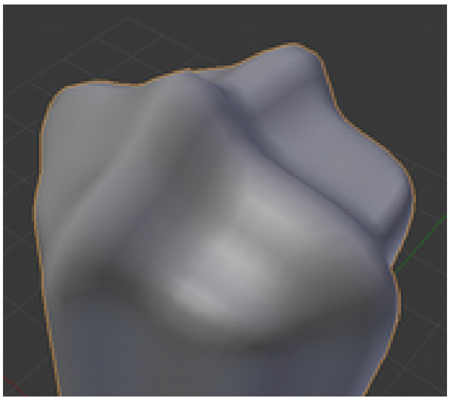


We learnt already how to copy objects (with CTRL+C/V), but now we also want to copy effects. Just press the copy button in the Wave modifier dialog.

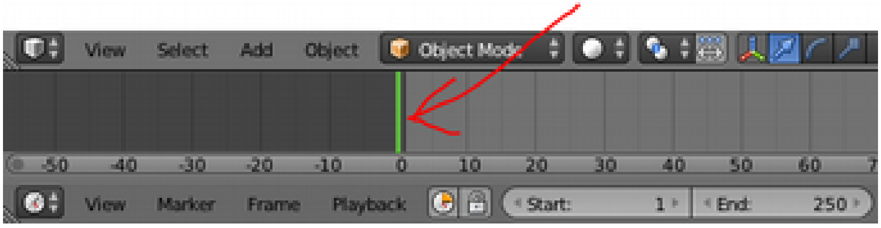
In this copied Wave modifier, you only have to change this option:

Motion:
☐ X
☑ Y
☑ Cyclic

From the top, the object should look like this now:



You have two waves running on the surface of the cell component. If you move now the time marker in the time line:

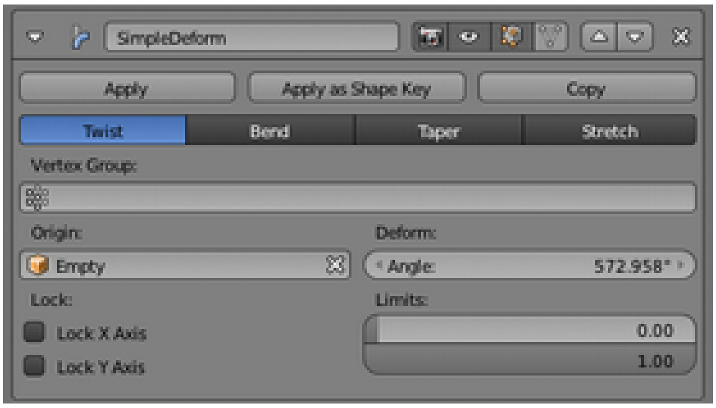


Start: 1
End: 250

You will see that the effect changes. In this tutorial, we are not interested in animations! But: we are interested in generating a nice surface. So you can use now the time slider to find a good combination of the different wave effects.

However, we want to examine another modifier. Let us still use the first time step. Now, add the

“Simple Deform” modifier and change the settings like this:



Vertex Group:

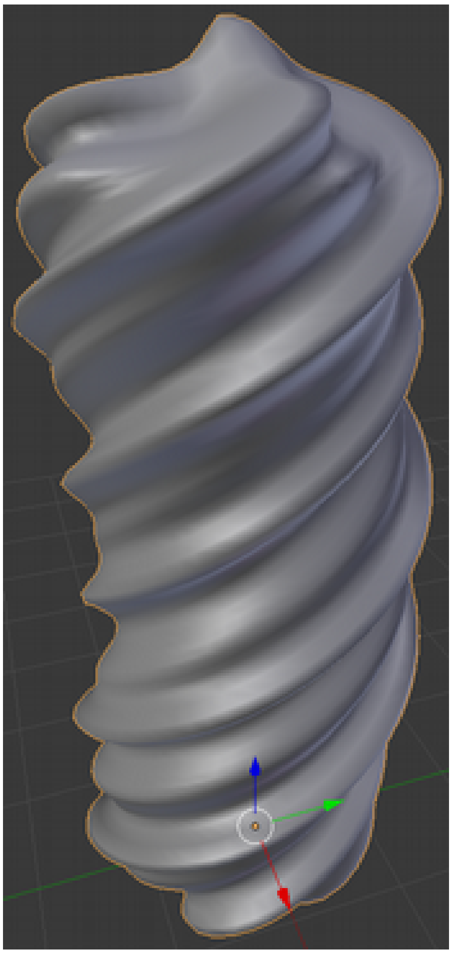
Origin: Empty

Deform: Angle: 572.958°

Lock:
☐ Lock X Axis
☐ Lock Y Axis

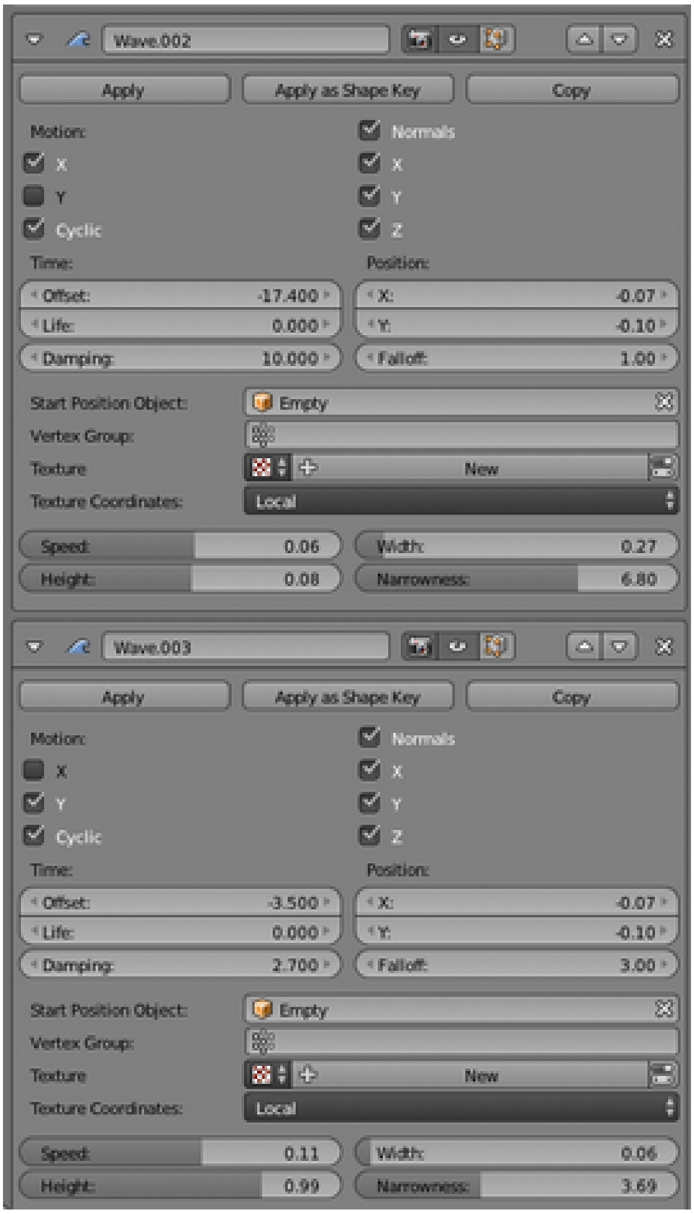
Limits: 0.00 1.00

Now your structure should look like this:



So we have some irregular waves now. By the way, you can always test different combinations of the modifiers by deactivating the modifier by clicking on the eye:

Optionally, you might want to add more Wave modifiers to gradually distort the surface of the cell component, e.g. with two additional ones:



Wave.002

Motion:
☑ X
☐ Y
☑ Cyclic
☑ Normals
☑ X
☑ Y
☑ Z

Time:
Offset: -17.400
Life: 0.000
Damping: 10.000

Position:
X: -0.07
Y: -0.10
Falloff: 1.00

Start Position Object: Empty
Vertex Group:
Texture:
Texture Coordinates: Local

Speed: 0.06
Height: 0.08
Width: 0.27
Narrowness: 6.80

Wave.003

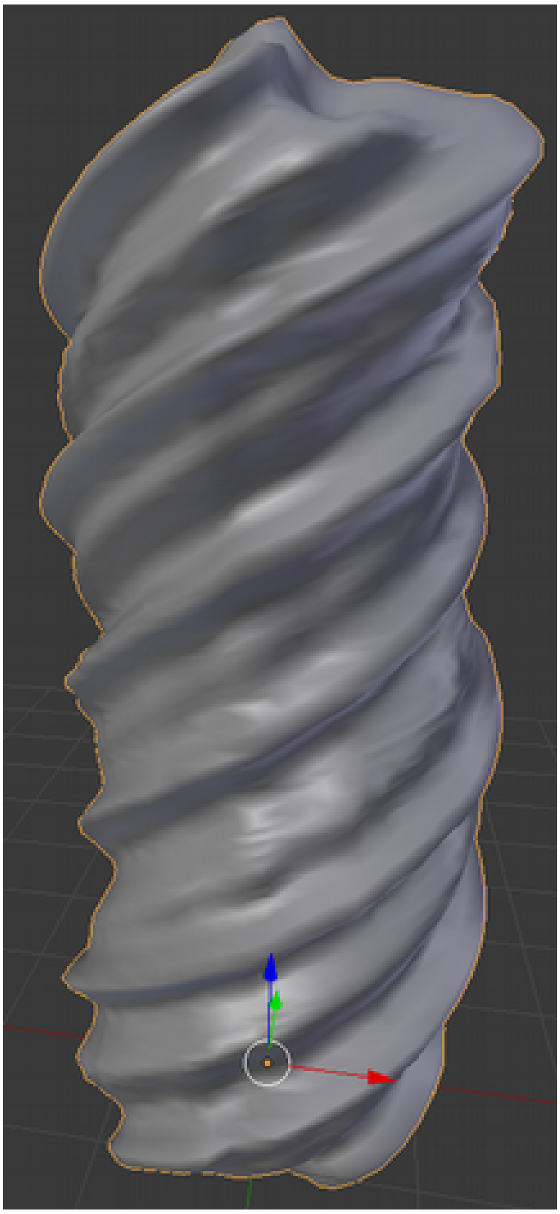
Motion:
☐ X
☑ Y
☑ Cyclic
☑ Normals
☑ X
☑ Y
☑ Z

Time:
Offset: -3.500
Life: 0.000
Damping: 2.700

Position:
X: -0.07
Y: -0.10
Falloff: 3.00

Start Position Object: Empty
Vertex Group:
Texture:
Texture Coordinates: Local

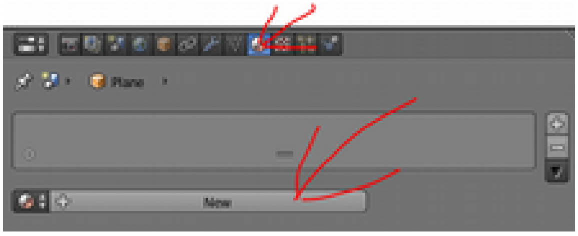
Speed: 0.11
Height: 0.99
Width: 0.06
Narrowness: 3.69



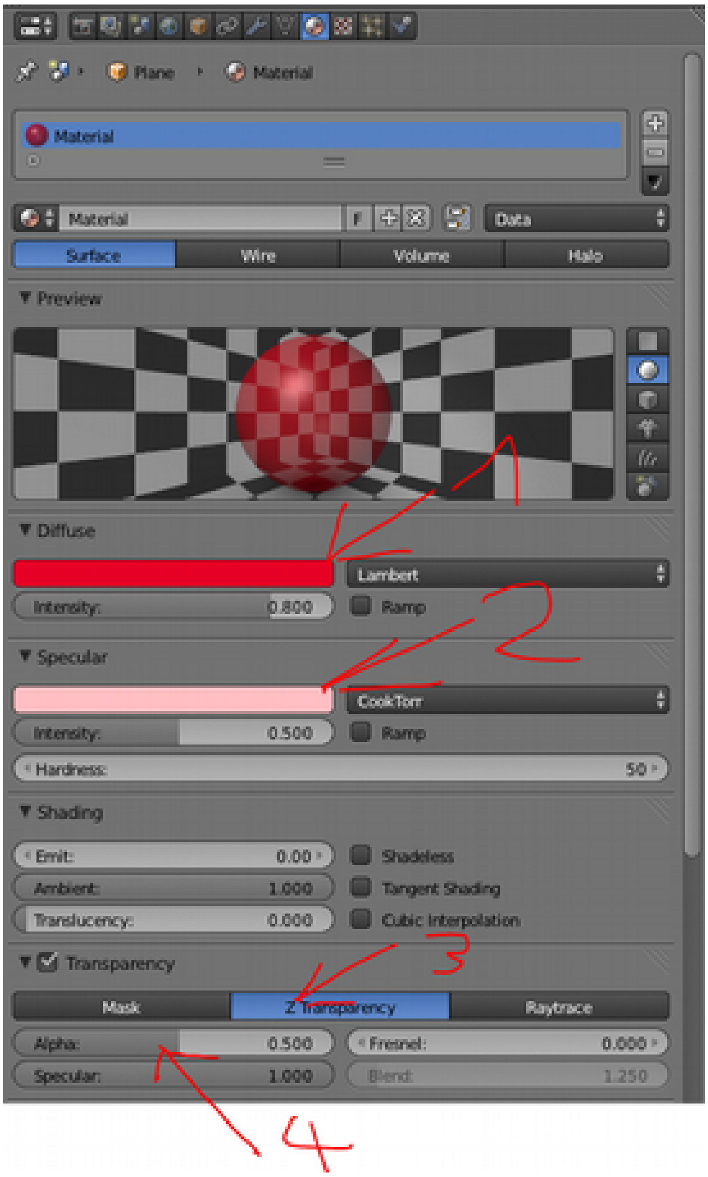
By changing now the time slider, you can change the structure, or by moving the Empty object to different positions nearby the cell component.

Blender: Add a Material to the Cell Component

Before we start to export our object, first, let us add a material. Otherwise, the object will be exported just white and the 3D structure will be hardly recognizable.



Now, press the “New” button and change the material like this:



Intensity: 0.800

Intensity: 0.500

Hardness: 50

Emit: 0.00

Ambient: 1.000

Translucency: 0.000

☑ Transparency

Alpha: 0.500

Specular: 1.000

Fresnel: 0.000

Blend: 1.250

1. Add a diffuse color:
the standard color of the object,

2. Add a specular color:
the color reflected at the bright spots where light hits the surface of the cell component,

3. Activate the transparency, and

4. Set the level of the transparency:
alpha=0 → invisible,
alpha=1 → fully visible.

Now, our material is finished.

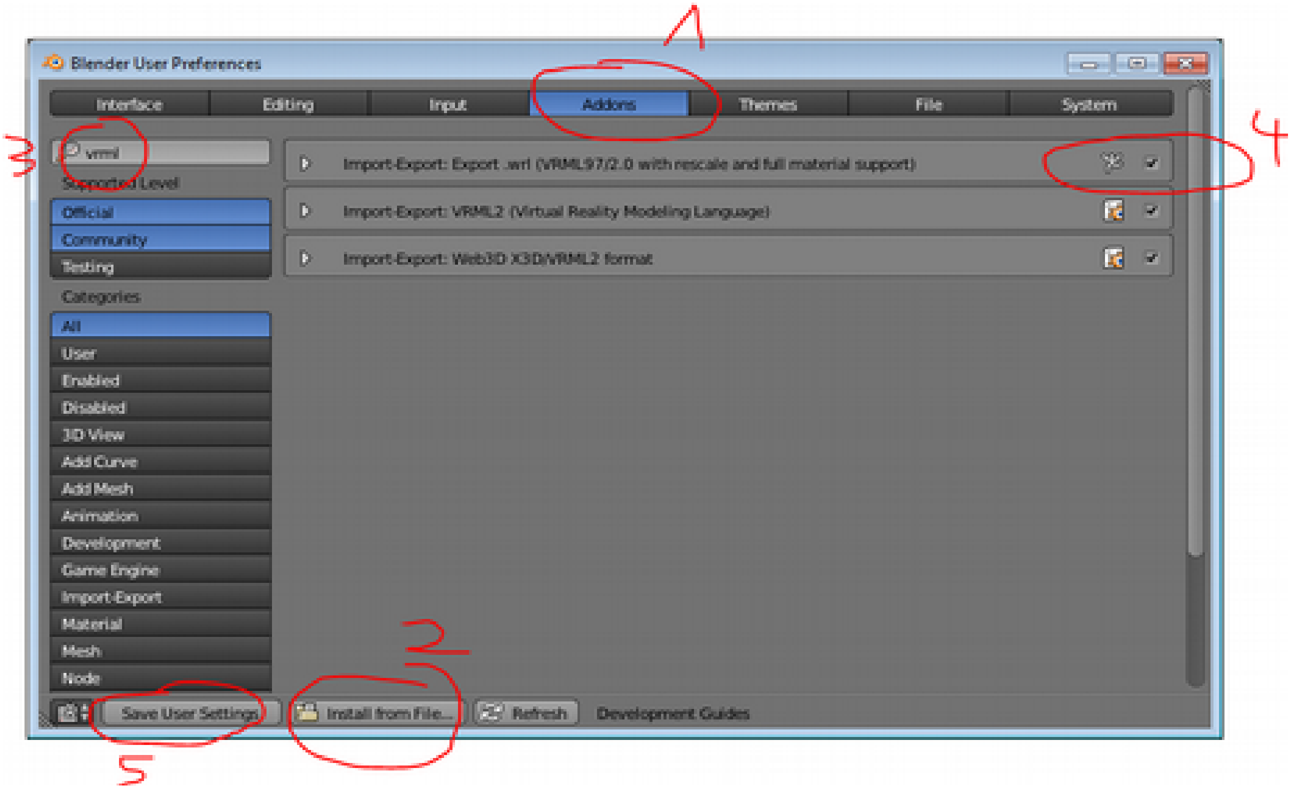
Blender: Export VRML97/2.0 for the CellExplorer

Now, we have created our cell model and assigned a material. To use it with the CellExplorer, it has to be exported to VRML2 format.

For this purpose, you have to first import and activate the VRML2 extension. In the already mentioned zip file mentioned above, there is the file: *vrml_export_266.py.* To get the full support of all functions described here, download the file *vrml_export_270.zip* from

http://www.cellmicrocosmos.org/Cmforum/viewtopic.php?f=21&t=750

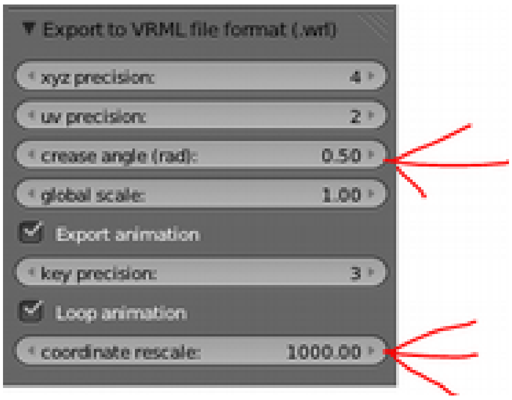
and extract it to your file system. Then, import it to Blender. First, open the “File” → “User Preferences”, then, 1. change to the section “Addons”, 2. click “Install from File ...” and select the previously extracted *vrml_export_270.py*, then, 3. search for “vrml” in the search box and finally, 4. activate the imported plugin. And 5., it is a good idea to press “Save User Settings”. Otherwise, you will have to repeat this whole process again the next time you start blender again.



Now, the VRML2 exporter is activated. Let us come back to our Blender cell component model:

Make sure, it is selected, because this VRML exporter exports only the actually selected shapes.

Go now to the menu and select “File” → “Export” → “Export to VRML (.wrl)” and find the following menu on the bottom left side:



Export to VRML file format (.wrl)

xyz precision: 4

uv precision: 2

crease angle (rad): 0.50

global scale: 1.00

☑ Export animation

key precision: 3

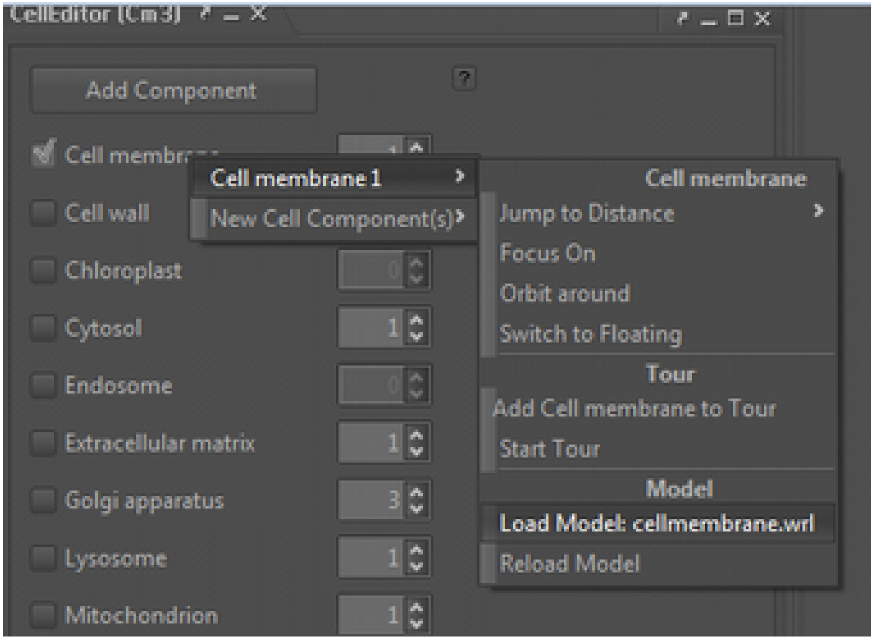
☑ Loop animation

coordinate rescale: 1000.00

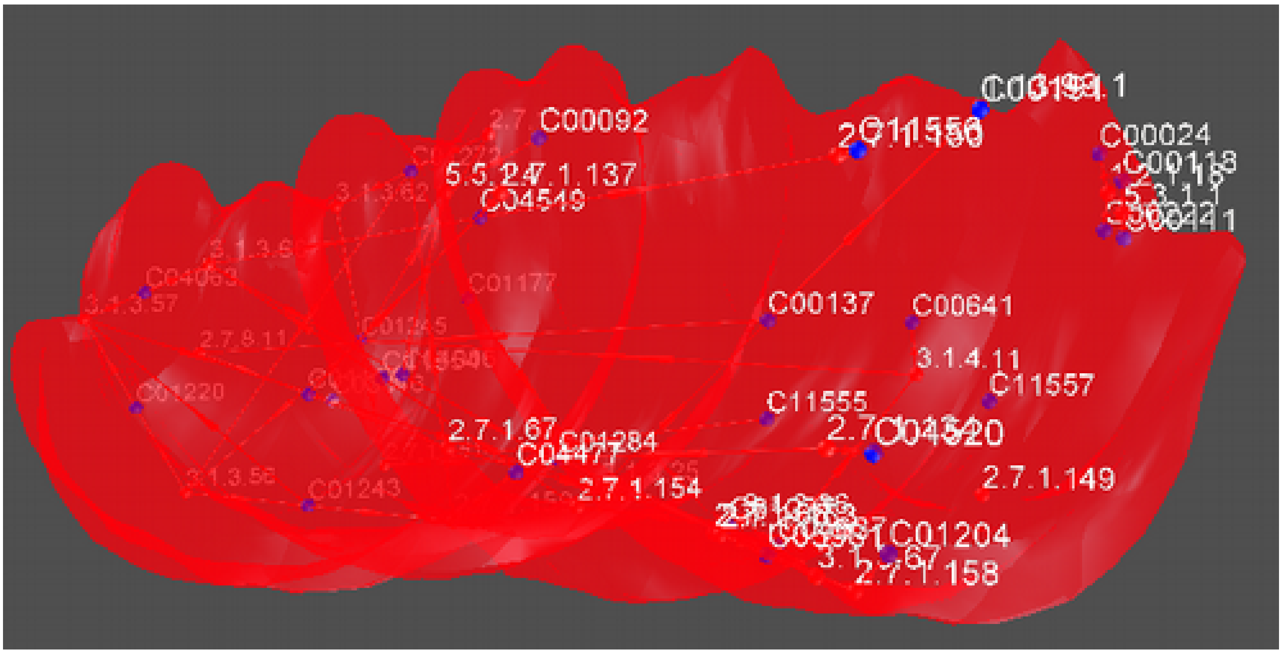
This nice plugin originally written by Sebastian Lieberknecht supports the following functions (*import functions are written in italic*):

- xyz precision: the number of digits behind the coordinates' comma, so here: XXX.XXXX
- uv precision: the number of digits behind the uv coordinates' comma, so here: XXX.XXXX
- *cease angle (rad): the cease angle which is relevant for the normal generation. This number must be larger than 0 if normals should be generated! Set e.g. the value to 0.5 and the 3D meshes will appear later smooth in the CellExplorer.*
- global scale: Change the global scale number for the whole VRML file. This number is interpret by many VRML viewers. However, the CellExplorer ignores this values.
- Export animation: if you want to export an animation, you can do this here. The CellExplorer ignores this value.
- key precision: relevant only for animations. How many digits the key should use.
- Loop animation: restart the animations
- *coordinate rescale: rescale all 3D shapes by using this number [version 2.70 only] use factor 1000 to export to the CellExplorer*

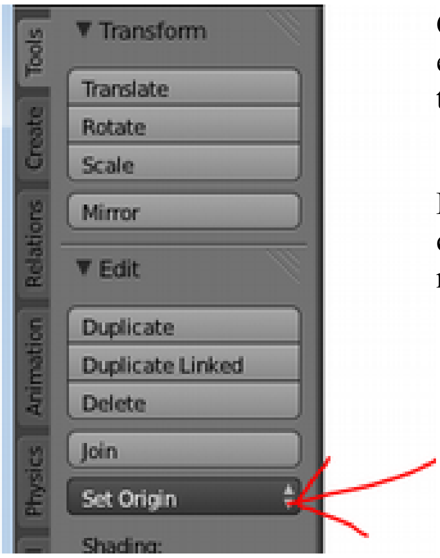
Right click on the term “Cell membrane 1” in the CellEditor window in the CellExplorer and select then “Load Model: cellmembrane.wrl” and select then the newly exported VRML file.



This image here shows the newly exported membrane in the CellExplorer:



An important aspect of the export from Blender to CellExplorer is that the origin of the exported shape(s) is in the center of the shape. This is especially important for the node placing algorithms in the CellExplorer, because they start the distribution process based on this center point. It is simple to correct the center point:



Click on “Set Origin” and set it to “Origin to Geometry”. Then, export again and reload the model by pressing “Reload model” on the pop up.

Finally, some words about extended cell components. Many cell components require the use of multiple layers. E.g., for the mitochondrion, 5 layers are needed:

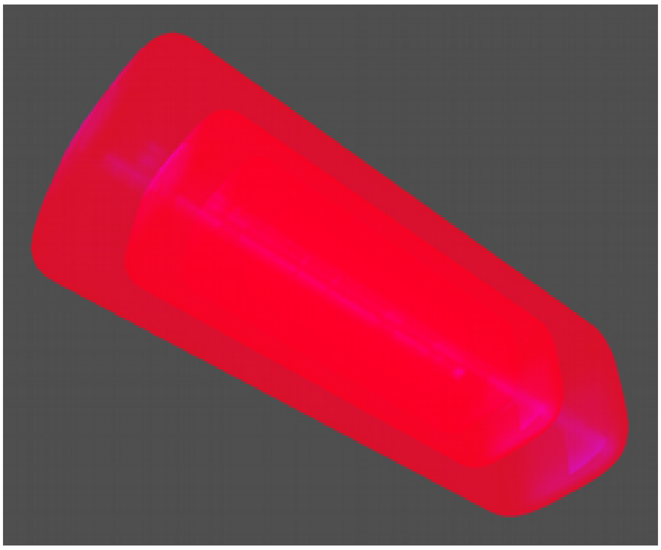
0. matrix
1. inner membrane
2. intermembrane space
3. outer membrane
4. cloud

Therefore, 5 layers have to be modeled in Blender. Before the model will be exported, it should look like this in the Outliner:

The use of numbering is a simple way to define the ordering. Obviously, we are starting from the internal membrane #0, the one, which is closest to the origin, and approaching then step by step to the outermost layer, #4. This ordering is required by the CellExplorer for the identifiction of the different shapes.

Now, make sure that all 5 layers are selected in Blender (you can also select them with SHIFT+LMB in the Outliner) and select again the VRML export function. After the export you will see that all layers are ordered as you previously defined it in Blender. *This works only with plugin version 2.70!*

This is, how a simple mitochondrion model is shown in the CellExplorer after export with a rescale factor of 100:



The alpha value of the transparency of the matrix (layer #0) and the cloud (layer #4) is set to zero.

Blender: Import a Cell Model from the CellExplorer

The following chapter will probably be not relevant for the CmCM project SS2014. But who is interested in this topic, here it is described anyway:

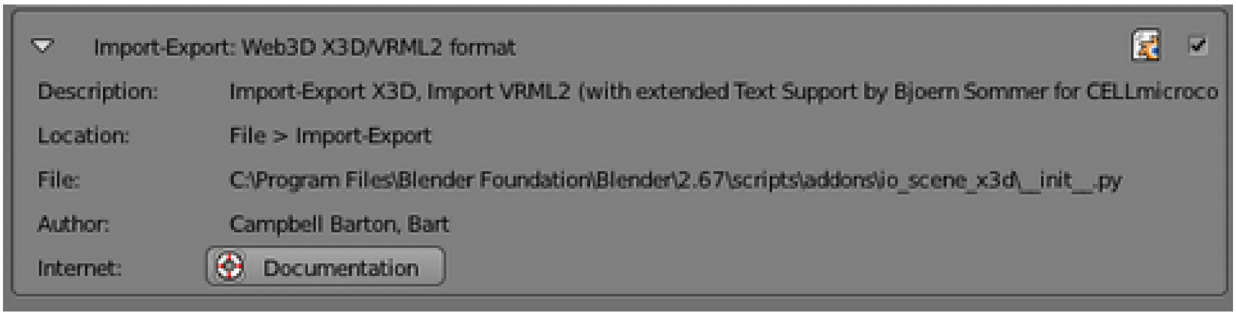
If you want to export a model from the CellExplorer to be used in Blender, export a cell model to a VRML file by clicking Save as VRML and save the file.

Now, we want to import the exported VRML file to Blender. For this purpose, make sure that you have substituted the folder io_scene_x3d in the Blender\2.67\scripts folder with the one downloaded from the forum website. This version support the import of text from VRML to Blender.

Now, start Blender. Go to

File → User Preferences

and make sure that the Web 3D X3D/VRML2 exporter/importer is activated here:

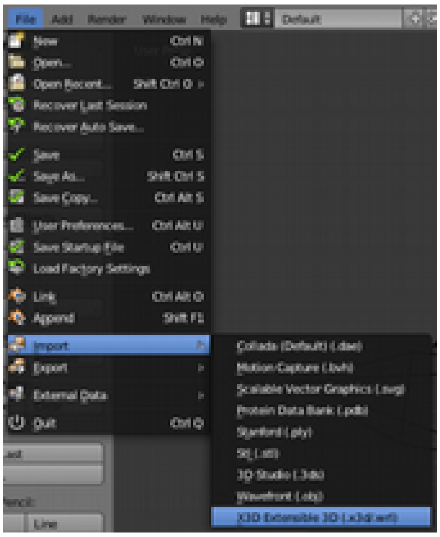


Import-Export: Web3D X3D/VRML2 format ☑

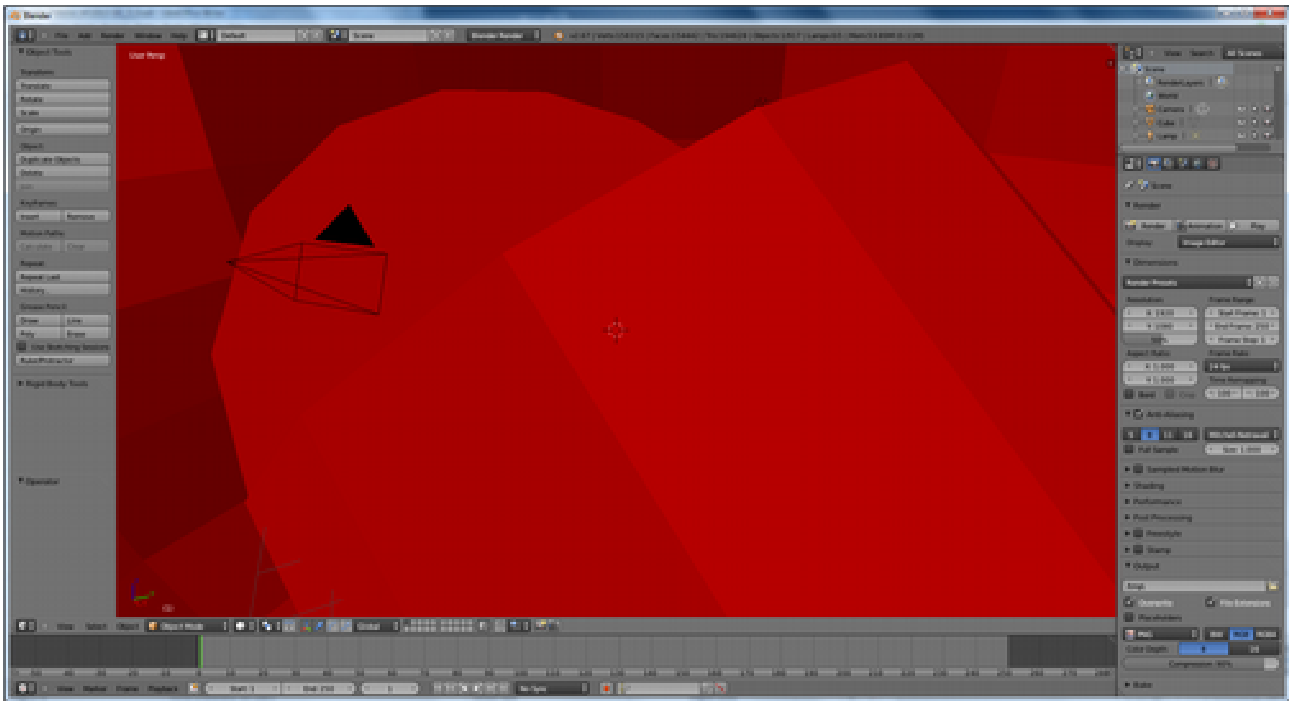
Description:	Import-Export X3D, Import VRML2 (with extended Text Support by Bjoern Sommer for CELLmicroco
Location:	File > Import-Export
File:	C:\Program Files\Blender Foundation\Blender\2.67\scripts\addons\io_scene_x3d__init__.py
Author:	Campbell Barton, Bart
Internet:	Documentation

Note: this plugin supports only the import of VRML files. If you want to export VRML files, use the file “vrml_export_266.py”, activate it just like the importer and find it in the Files → Export dialog. This exporter can also be used to prepare shapes for cell components for the CellExplorer.

Now, import the .wrl file by using:



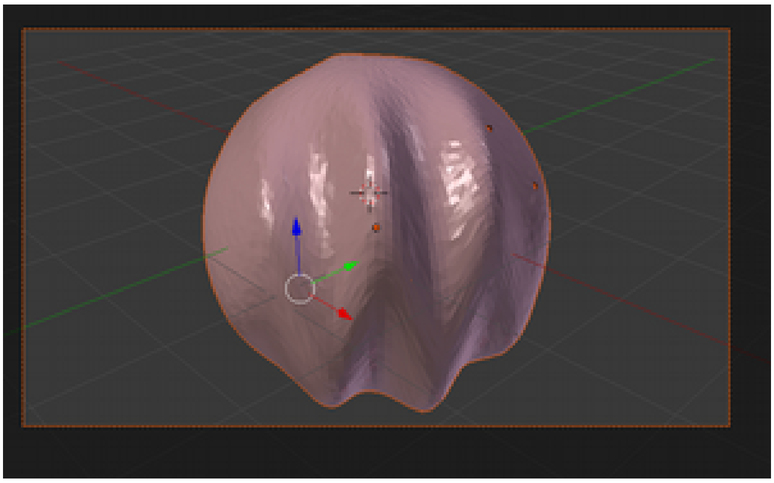
After the file is important, you will see something like this:



Obviously, the 3D model is much too huge.

Select the whole model with *A* and then use now a combination of scaling (with *S*) and grabbing (with *G*) to decrease the size and to change the position of the cell model. Combine these two modes with restricting the movement to a single axis, e.g. by pressing *Z* afterwards. Always make sure that the cell model does not disappear from the view port and move it towards the center of the membrane by looking from the top or from the side. A good trick to take the line into account which appears during pressing *S*. This line indicates the direction and the position of the rpugh center of all shapes.

In the end, the cell model should be located in the center:



You know how to handle this model from the previous tutorials. One last hint should be given. If you look to the Outliner, you will see a lot of different shapes. The problem is now that it is not easy to identify the correct cell component in the Outliner. Just select now the visible cell components and give them names. But how to do this?

Just change to “Selected” mode in the Outliner and click the cell membrane and change the name of the shape with a name like “ShapeIndexedFaceSet” to “Cell Membrane”:

Deactivate the eye symbol and hide the cell membrane. Select the mitochondrion component and change then its name.

Now, you can filter objects in the Outliner. Go back to “All Scenes” and use the search field to search for a cell component name.