

PROTEIN EXPRESSIONS

Study N. 2

The video Protein Expressions. Study N. 2 is a re-elaboration of our first video, released few months earlier (April 2009). The following text also covers most of the description prepared in that occasion, with some additional information, and including the new scenes introduced.

More technical and/or scientific details can be found on our website, and will be submitted for publications to the scientific literature.

What follows is the description of some of the technical and scientific aspects of the video, scene by scene. We are aware of the fact that the terminology used can be difficult for non- professional biologists or graphic experts. Hopefully the interested reader will take occasion to get some insights from many excellent cell biology and 3D computer graphics textbooks.

1. Red Blood Cells with a White Blood Cell (T cell)

The set is a fine capillary. It is populated by red blood cells, platelets and a few white blood cells. The scene was built in Blender. The capillary is a cylinder on whose surface (inside normals) is mapped a microscopic photograph (manually made seamless) of epithelial cells grown to confluence in a Petri dish. The number of cells in the lumen of the capillary is lower than it is in reality: they are so crowded that it would be impossible to see in between them. All objects (red cells, white and platelets) are present in dimensional and numerical proportion.

Red cells are all instances of a simple polygonal sphere deformed to become biconcave; these instances are set as rigid body objects and sent through the vein (positioned vertically) by gravity. The Blender Game Engine evaluates collision between cells and with the capillary walls.

The white cell, is a mesh modelled by extrusions and moved by means of both a skeletal system and an animated displacement texture. It moves along a curve path, similar to the camera.

2. Flight over cell surface.

The membrane was obtained imposing properties derived from membrane biochemistry to several particle systems (in Maya): single lipids, groups of lipids (rafts) and groups of proteins distributed on the surface. This was converted to a grey scale texture attributed to the displacement node of the surface using Blender SuperTexture Node and Texture Plug-in.

The objects flying with us over the surface are modelled on the largest objects (besides cells and platelets) found in human plasma: High and Low Density Lipo-proteins (also known as good and bad cholesterol), and IgMs, large multiprotein complexes that look like 'snow-flakes'.

They are added to the scene as polygonal meshes in 3 different depth layers (each one containing few thousands objects). Their motion was calculated using spherical bounding boxes actors moved around with random impulses via Python.

3. The membrane.

Based on the texture above, the rafts were constructed as a Game Engine structure in Blender, with proteins loosely connected with rigid body joints. Proteins in the raft were modelled (at low resolution) on the basis of pdb files: CD4 (1WIO), Thy-1/CD59 (2UX2), TCR (1KCG), CD3 trimer (1XIW, 1SY6, 1SY6), Toll Like Receptor (2Z7X), ICAM-1 (1IAM for domains 1 and 2, and 1P53 for domains 3, 4 and 5).

The group of proteins was set to follow the centre of a raft. This motion was recorded and provided the 2D surface sliding of proteins on the membrane. The Z (vertical) motion to follow the wavy surface was obtained by dropping the proteins at each frame on the distorted plane. An additional rotatory motion was attributed to each protein with a random generator.

Each protein also has a skeleton that provides the inner motion. Also oligosaccharides chains are modelled from pdb structures and moved with an internal skeleton by another random motion generator, faster and with wider angles

relative to proteins.

The Potassium channel is modelled from the structure in pdb file 1LNQ (side chains added with Swiss PDBviewer).

4. Microtubules.

Once inside the cell (after falling into the K channel), we meet a bundle of microtubules. These are built with an array of tubulin dimers (pdb file 1TUB) displaced by the appropriate distance to form a 13 filaments ring. The ring is repeated (another array) to form the pipe.

Running over the central microtubule we see KIP3, one of the many proteins that populate the cytoskeletal structures in the cell. Its shape is modelled on the pdb 3KIN, and its motion was obtained by a system of bones (moved with IK and using stride bone) based on the motion of Kinesin, as described in the scientific literature.

Other proteins in the scene are NFkB, Diaphanous (1Z2C) and Aurora (2BFX), rendered according to our procedure and moved through a similar mechanism as the proteins over the cell surface (scene 2).

5. Calmodulin.

The movement of Calmodulin (CaM) is the most scientifically relevant part of the entire video. CaM is a small protein (148 aminoacids, 2262 atoms, excluding Hydrogen) structurally organized in two parts (heads, or globular domains) connected by a flexible linker. The two parts are capable of capturing Calcium ions, and when this happens the lobes open and become reactive towards other proteins, thus transducing the signal delivered by the Calcium wave.

We meet apoCaM (Calmodulin without Calcium) after the trip along the microtubules. Protein movements were obtained elaborating on an NMR collection (1CFC) using the Game Engine of Blender according to a flow as follows:

all atoms of the protein (excluding hydrogens) are imported in Blender (we wrote specific scripts to import and export pdb files) in two different positions, chosen as the farthest apart as RMSD, separated by X frames (say 500). Atoms are connected with rigid body joints (chemical bonds). Blender Game engine interpolates between conformations evaluating joints and collisions. Intermediate conformations so generated are automatically adjusted using the GROMOS force field implemented in Swiss-PDBviewer and compared (RMSD) with other conformations derived from the original NMR collection. Any NMR conformation closer than 2 Å is imposed as intermediate keyframe in the correspondent time between the initial positions, and the process is repeated.

With a few cycles of this procedure we have generated a continuous motion that covers a geometrical distance of 21 Å in steps of <1 Å, 'visiting' a dozen conformations experimentally derived. These are set in Blender at 50 frames distance, and again interpolated by the GE. Finally, each frame of the global movement is exported as a pseudo-pdb file and sent to biophysical programs that introduce some necessary minor adjustments to the position of atoms and reintroduces hydrogens.

For visualization of chemical and physical properties, we have developed a visual code applied to the surface of the molecule without making use of colors.

Each frame from the animation (pdb file), is converted to .pqr (a program that introduces proper charge to every atom) and elaborated with a series of libraries and other programs and scripts (APBS and pyMLP.py) that calculate electrostatic and lipophilic potential (EP and MLP respectively) values. Using a home-made program, these fields are mapped on the protein surface (SAS, Solvent Accessible Surface, obtained as an .obj file from PyMOL) and associated to all vertices of the mesh for LP or transformed in a series of lines with a direction (for EP). The values of MLP are fed to the texture nodes of Blender and are finally expressed as scale of smooth-shiny to bumpy-dull. The values of EP and represented as particles running along the filed lines, from positive to negative.

While we observe Calmodulin idling in its apo form, a Calcium wave is shown, which induces a major conformational change. This was obtained by morphing the protein from Conformation 21 of 1CFC to 11 of 1X02, which collects a number of conformations in the Calcium bound form. Also in this case the transitions were obtained using Blender as well as some scientific applications.

6. MLCK and the plasma membrane.

In the following scene we meet a new protein : Myosin Light Chain Kinase (MLCK). This very large protein is regulated by CaCaM: it sits in the vicinity of the contractile ring (described below) and its role is to activate the myosin movement by adding a Phosphate group to one of its subunits. Because all movements need to be coordinated, they are mediated by CaCaM: the signal arrives in the form of Calcium, which binds and activate

Calmodulin, which now can go and activate MLCK (as well as other components, not visible here). The only part of MLCK modelled and moved at the atomic scale is the large active domain in the front. MLCK tail is modelled on the basis of atomic information, but is moved (by hand, as scientific information is not sufficient here) using normal bones accurately placed within the structure.

The scene is set during the cytokinesis, when the contractile ring pulls the membrane that will eventually divide the cell in two. Here a bundle of actin filaments are pulled by myosin, as described in the next section

The underlying membrane is plane converted to cloth, contracting along the middle line through pinning to a cylinder scaling in time, and covered with 50.000 particles (hair) that follow the plane geometry to simulate presence of many proteins.

7. The contractile ring

To create the actin filaments, we positioned the actin monomers as described in the PDB file 1ALM (crystal structures). This model, containing 5 monomers, was imported in the Blender scene via Chimera. Since the models in this file only contain the alpha-carbon atoms, another, atomic resolution crystal structure of actin (1ATN) was used to obtain a high resolution actin filament model. In Blender an array modifier was created, to multiply the single actin model by values derived by carefully adjusting the offset and rotation settings of the monomers in the crystal. The array-based actin filament can be elongated in a snap.

Myosin filaments were created using of PDB files 1I84 and 1SR6, (two extreme conformations of the heavy chain motor domain) and 1NKN (coiled-coil tail structure from two intertwined myosins), which were loaded into Chimera, together with 1ALM (5 aligned actin monomers). The latter file also contains a myosin head domain bound to the actin filament. This gave us the opportunity to align the myosin head group to the actin filament. The other myosin structures were aligned to the 1ALM file and the surface models were created and exported as described. The models also include the myosin essential light chain and regulatory light chain molecules, wrapped around the neck domain of the myosin heavy chain molecule. Once imported into Blender, the myosin motor domain movement was animated using an armature for the pre-power stroke model. The armature was connected and subsequently posed to closely match the myosin surface model in the power stroke position. Care was taken to not only animate the large displacement of the head group, but also the more subtle displacements of smaller subdomains of the proteins, through an additional armature into the bumps of the protein surface model. The resulting rigged model could now be moved from the pre-power stroke conformation to the powerstroke conformation by moving from one pose of the armature to the other. The final movement of the myosin molecule is composed of five partly overlapping steps:

1- power stroke (first 20 frames); the motion of the myosin head while it is attached to an actin filament; this motion causes the displacement of the actin filament.

2- release and lift (frame 20-40). The myosin head is released from the actin filament and the head is lifted away from the actin.

3- prepare for power stroke (frame 29-55). The myosin head is rotated around the neck to reposition it for the next power stroke.

4- drop (frame 41-66). The myosin head moves towards the actin filament.

5- connect (frame 66-67). The myosin again grabs an actin filament; to visualize this, the motion of the bones stops for just one frame.

The tail domains of two myosin heavy chains twist around each other to form a coiled coil. The 1NKN PDB file was used to model such a coiled coil. This model, however, only contains 89 amino acids per strand, whereas the tail domain of a myosin heavy chain molecule extends over hundreds of amino acids. Therefore, the model was imported into Blender and multiple copies of the small model were carefully aligned and connected manually until the correct tail length was reached. Myosin is active as bipolar filaments, which are composed of 10-20 pairs of myosin molecules. To mimic this, a bundle of myosin tails was created by joining a number of tail models together. Twenty myosin head models were placed pairwise at the two ends of this filament model. The conformational movement of the headgroups was subsequently shifted with respect to each other to create a more realistic, shuffled animation. Finally, the acceleration of the movement by Calcium waves that lead to MLCK activation had to be animated. For this, each armature object (which drive the movement of the myosin head groups) was given a so-called time ipo. This allows for acceleration or deceleration of the movement of the head groups. As all the head groups were given the same time ipo, the speed of movement can be globally changed easily.

OTHER NOTES

Time Considerations. The speed at which biological activities are performed is such that nothing could be followed by human eyes. In this video we make no attempt to follow timing considerations.

Music and Sound. Are molecules silent? Or do they produce sound? Just as the scenes we show are, in reality, invisible, they produce no sound audible to the human ear. The music accompanying the video was created by Antonio Dell'Aquila, a friend of the lab, a talented artist and musician who interpreted the images according to his taste and art.

Why no colors? Is inner life all black and white? Please ask another question :-)

PRINCIPAL AUTHORS:

Monica Zoppè	Scientist, IFC – CNR, Pisa. <i>Corresponding author</i>
Raluca Andrei	PhD student, Scuola Normale Superiore and IFC – CNR, Pisa
Stefano Cianchetta	3D graphic, IFC – CNR, Pisa
Tiziana Loni	3D graphic, Big Bang Solutions, Navacchio (Pisa)
Maria Francesca Zini	Scientist, IFC – CNR, Pisa
Yuri Porozov	Scientist, IFC – CNR, Pisa
Leonard Bosgraaf	Dept. of Biochemistry, Univ. of Groningen, Netherland
Marco Callieri	Scientist, Visual Computing Lab, ISTI – CNR, Pisa

PROGRAMS USED

Graphics and image processing:

Blender www.blender.org
Maya Autodesk <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=7635018>
The Gimp www.gimp.org
Djv_view http://djv.sourceforge.net/djv_view.html
ImageMagick www.imagemagick.org

Scientific

VMD <http://www.ks.uiuc.edu/Research/vmd/>
Swiss-PDBviewer (Gromos43B1) <http://spdbv.vital-it.ch/>
Chimera <http://www.cgl.ucsf.edu/chimera/>
Reduce - MolProbity <http://molprobity.biochem.duke.edu/>
PDB2PQR <http://pdb2pqr.sourceforge.net/>
PyMLP.py <http://code.google.com/p/pymlp/>
APBS: <http://sourceforge.net/projects/apbs>
Several home made scripts and programs

Other References

Protein Data Bank <http://www.pdb.org>
Antonio Dell'Aquila (music) <http://www.antoniodellaquila.com/>

We have used wherever possible open source programs. All our scripts and programs will be packaged in user friendly form, and made available in open format. Some are already available for download from our website. For more information, or comments, please email us mzoppe-at-ifc.cnr.it

Visit our website to stay updated with the latest news www.scivis.ifc.cnr.it