Chapter 11
The Representation of Electrostatics for Biological Molecules

Monica Zoppè and Tiziana Loni

Abstract Biological molecules live in an environment in which most of the forces that determine their activity are (at least apparently) different from those that guide the world visible to humans. These forces can be roughly classified as Brownian motion, lipophilic/hydrophilic interactions, and electrostatic potential. In the study and representation of proteins and other biological molecules, and especially their activity and interaction, it is therefore extremely important to be able to observe these forces in a meaningful way. This will lead to better understanding of dynamic interactions, and for a greater capacity for description and manipulation. While the calculation of these forces typically provides numerical data, it is not always easy and intuitive to have them represented in a way that makes sense to human beings, whose visual intelligence is one of the most highly developed. In the past few years, we have developed BioBlender, a software program dedicated to the intuitive visualization of proteins, their properties, and their interactions with other proteins, other biological molecules, and the cellular environment. BioBlender is based on Blender, one of the most powerful engines of Computer Graphics (CG) for 3D content management, i.e., creation, animation, texturing, and rendering of visual objects and scenes. Blender is the only complete program dedicated to 3D that is open source, a feature important for users who want or need to introduce new features, in our case the ability to handle biological objects starting from files in PDB format. In this chapter, we will consider some issues of molecular visualization, and describe some of the features of BioBlender, with particular focus on the calculation and rendering of Electrostatic Potential (EP).

11.1 Introduction

The study of biological entities, before and after the advent of microscopes and other technical instrumentation, has always relied on visual aids, such as the well-known anatomical drawings of Leonardo da Vinci and Vesalius [1, 2]. As soon as

M. Zoppè (✉) · T. Loni
Scientific Visualization Unit IFC - CNR, Pisa, Italy
e-mail: mzoppe@ifc.cnr.it

W. Rocchia and M. Spagnuolo (eds.), *Computational Electrostatics for Biological Applications*, DOI 10.1007/978-3-319-12211-3_11

the first information on molecular structures became available, the need for visual representation was recognized, and several different techniques were developed to provide 2D images and 3D objects or representations, capable of showing the general shape and other biological details [3, 4]. Visual observation of objects and images has greatly helped in the understanding of function and in the discovery of new important features, such as in the very famous case of DNA, whose structure was revealed while "playing" with physical models of the four bases A, T, C, and G [5].

Nowadays, most representations are built using the capabilities of Computer Graphics (CG), one of the branches of computer science that has advanced greatly in the last years, thanks to the input and requests from the entertainment industries of cinema, TV, and computer games.

11.2 Computer Graphics in Structural Biology

It is possible for structural biologists to utilize the advances of CG to represent proteins and other macromolecules, with the best possible accuracy and fidelity to the experimental data. Many of the most common instruments developed for visualizing structural data have been built by and for the scientific community: VMD [6], PyMOL [7], Swiss-PDBviewer (spdbv) [8], and others are dedicated programs that incorporate some features of CG, but are essentially scientific tools, with several dedicated functions including chemical and physical analysis instruments. For example, they can build new chemical bonds, add or remove H atoms according to pH, they can calculate energy for different states, align proteins that share a degree of similarity, etc. These tools can also produce high-quality renderings, offering some control over lighting, shadow casting, coloring and texturing to some degree; indeed, many scientists do use them to produce figures of great interest.

A different case is the adaptation of highly sophisticated, professional-quality CG programs to the representation of molecular life. This possibility is being explored by our group and, to our knowledge, by three others: the Boston group that developed Molecular Maya [9] and a 'Molecular Flipbook' [10], a Californian group, led by Art Olson which is engaged in the development of ePMV [11], and the Australian group of Drew Berry [12], that has produced very beautiful molecular animations using sophisticated 3D techniques, although it has not developed specific instruments.

The attitude and philosophy of the different groups is reflected in the variety of solutions proposed for describing different aspects of molecular life. For example, the Molecular Flipbook enables scientists to visualize molecular models providing some simplified options derived form the vast array of 3D manipulation techniques. On the other hand, ePMV is designed with a special attention to the simulation of spatial arrangement of different macromolecules and organelles in cells. In contrast, the major goal of our effort, is aimed at visually representing objects and forces in the cellular environment.

Altogether, all the efforts of the new discipline of molecular visualization, provide a choice of tools and techniques, that might well serve to the better understanding of molecular and cellular biology, while producing visual products (images and movies) that stimulate both scientists and the public alike.

In the rest of this chapter, we illustrate how many features of the CG repertoire can be dedicated to showing properties of proteins and other biomolecules not just as a medium for communication among scientists, but as a way to explore the intricacies of the very complex activities that take place in cells, with several entities of various nature (proteins, lipid membranes, sugars, small molecules) interacting at high speed and in a coordinated way. To show this type of activity, we not only need to build the 3D structure of the objects involved and their movements, but also want to deliver information on the forces and environmental conditions that influence or determine their behaviors.

For this reason, we have used Blender [13], a major CG instrument, which incorporates a wide collection of CG options, and is distribute as an open source program. Using this instrument, we are working on the visual codification of concepts which are not in our human immediate experience, such as lipophilic and electrostatic potentials, pH and reducing power.

While providing visual clues for new concepts, we also aim to show a world in which some of the most natural experiences are not found or are negligible, such as gravity or light itself. It is clearly an exercise of compromise; for example, light is a *conditio sine qua* we are not able to visually perceive the presence of objects; at the same time, the use of colored light can be exploited to transmit information about "invisible" features. One aspect of the necessary compromise is the obligatory choice of removing water from the visible objects, although it is often important for molecular activity. If we were to represent water in atomistic detail, we would face a wall of water molecules, with some occasional objects, and would completely miss the depth of view offered by the 3-dimensional environment that we are so interested in watching.

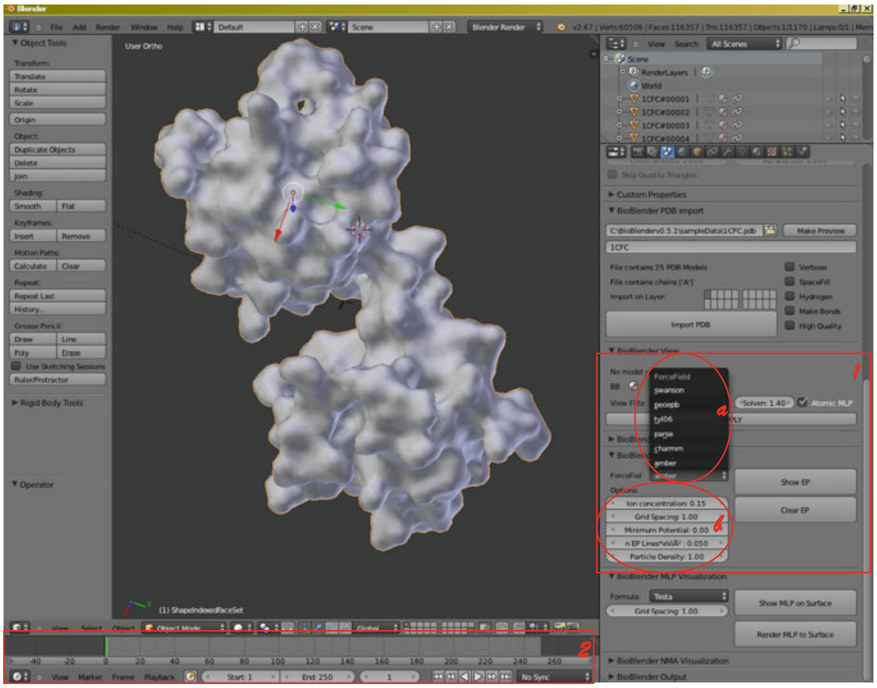
In this chapter, we focus on studies devoted to the representation of Electrostatic Potential, with a short introduction to Blender, BioBlender, and some of their relevant aspects.

11.3 Blender and BioBlender

Blender [13], the package developed by the Blender Foundation, is an extremely powerful and complex collection of features and functions dedicated to the creation and manipulation of 3D content. Its main environments can be roughly classified as Modeling, Animation, Texturing and Lighting, Rendering, and Video editing.

BioBlender [14] is an Add-on module scripted in Python, which also includes other programs necessary for biological, chemical, and physical computations (PyMOL [7], PDB2PQR [15], APBS [16], SciVis.exe [see below]). Figure 11.1 shows BioBlender interface, with some of its most relevant features.

Modeling. In CG, modeling refers to the creation of virtual objects in the 3D scene. These are defined as points, edges, and faces described by their xyz coordinates, and



C:\BioBlenderv0.5.1\sampleData\1CFC.pdb
1CFC
File contains 25 PDB Models
File contains chains ['A']
Import on Layer:
Solvent: 1.40
ForceField: amber
Options:
Ion concentration: 0.15
Grid Spacing: 1.00
Minimum Potential: 0.00
n EP Lines*eV/Å²: 0.050
Particle Density: 1.00
Formula: Testa
Grid Spacing: 1.00

Fig. 11.1 BioBlender. Screen-shot of BioBlender, showing (*1*) the input settings for EP, in which the Force field (*a*) and other parameters (*b*) are chosen by the user. Also shown are the main 3D viewport, with Calmodulin, and the timeline (*2*)

their final shape can be achieved by manual and/or procedural processes, whose description is beyond the scope of this publication. When we consider biological objects, they are typically composed of a (large) set of atoms with identity and position, bound to one another by known chemical bonds; these are typically stored in Protein Data Bank (PDB) files, which lists them in a format readable both by humans and by machines. To import biological objects in Blender, BioBlender provides a PDB parser that reads the PDB file line by line and builds the protein in the 3D space, using two libraries: an atom library (that contains the list of spheres of defined radius and color), and a library (including aminoacids, nucleic acids, some oligosaccharides, and a few other molecules of biological interest) in which chemical bonds are listed.

Thus, when a biological object is built in BioBlender, its chemical structure is also imported. The ultimate scope of BioBlender, however, is to present the activity and behavior of molecular objects, and rather than showing their internal atomic construction, we have elaborated a way to show the entire object, as the molecular surface (Solvent Accessible Surface) calculated by PyMOL. Secondary structures, internal cavities, and other features not exposed to the surface are not represented.

Animation is the process of CG that determines the change of a feature (not necessarily a mesh object) in time. Time in CG is managed as a series of "frames" (at least 24 per second), accessed via a timeline. The 24 frames originated in the early days of film projection; when shown 24 frames per second (fps) the human eye perceives a continuous flow, tricking the mind into seeing a smooth motion.

Any element in a scene can be changed in time, the location/rotation/size of an object, the position of a subset of its elements (points defining the mesh), the intensity of a light, or the size and color of a texture, to give a few examples. Users can define the state or value of the selected feature at two time points, and an internal engine interpolates between the two. The mode of interpolation can be selected (step, linear or Bezier spline). In some cases, special features allow for animation through more complex mechanisms: for example, a character can be equipped with internal bones, connected through joints, which are moved by *inverse kinematics* (IK).

Many biological molecules exert their function by means of an activity that includes conformational change, either through binding to another molecule, or via one or more active steps. The best-known examples are the motion of motor proteins (Myosin, Kinesin, Dynein), and the activity of transporter proteins (from the mitochondrial proton pump to the numerous active and passive transporters on all biological membranes). It is therefore of great interest to be able to visualize their motion. All atoms molecular dynamics (MD) is a very popular method for calculating molecular movements; however, it is very demanding in terms of computing power and time, and alternative methods have been proposed that reduce the need for enormous computations; these include "simplifying" molecules by reducing the number of components (as in Coarse Grain MD), and methods that extract information from a relatively short simulation time (Normal Mode and Principal Component Analysis). With BioBlender, we have proposed a new and even more simplified interpolation that uses the Blender Game Engine, as described in Zini et al. (Ref. [17]), to calculate the atomic motions of a protein, provided in at least two different conformations. Briefly, one or more molecules can be imported; each one of them can be imported in different conformations, at different time points of the time line (see below).

BioBlender builds molecules connecting covalently bonded atoms with links of the type "Rigid Body Joint." This kind of link allows only for rotation along the bond axis, and is set to be unbreakable, in order to reproduce in a coarse but reasonable way the kind of motion that underlies atomic motion. We reasoned that in very large and complex molecules, bond length extensions and planar angle changes would be minor, relative to changes in torsion angles that lead to major conformational changes.

BioBlender users can import any molecule in several different conformations, and can define the distance in time (expressed in number of frames) between the conformations. Since BioBlender calculates the conformation for every frame, setting 100 frames is equivalent to setting 100 steps in the transition between two conformations. This process is also called morphing.

With this setting, in order to interpolate the movements of all the atoms of a protein between the given conformations, BioBlender uses the Game Engine (GE) incorporated in the program which includes a physics engine based on the Bullet physics library [18]. The GE includes sensors, actuators, and controllers that can be employed to mimic the behavior of atoms. Each atom receives as input the next step in the direction toward the position in the final conformation; however, it is bound by the collision detector (if two atoms get too close, the trajectory is deviated) and by the kind of bond that links it to other atoms; therefore, the entire motion can be described as a series of torsions along atomic bonds.

The morphing is calculated in a number of steps decided by the user, and set as number of frames between subsequent conformations. The system has been validated using Calmodulin as a model protein [17].

Texturing in CG is the process used to provide an object with surface features that convey to the viewer indications of the object material; texturing includes the choice of color (hue, saturation and value), roughness, reflection, pattern, and so on.

The surface of proteins is calculated in BioBlender by PyMOL as the solvent accessible surface area (SASA), and imported in the 3D scene of Blender as a mesh.

We have elaborated a texturing code to represent the lipophilic/hydrophilic nature of the surface of molecules, whereby most lipophilic areas are represented as white, smooth, and reflective, and most hydrophilic areas are dark, rough, and dull. The values of lipophilicity are calculated in BioBlender on the basis of an Atom Type library and a series of formulas, and mapped onto the surface of the protein as described in [19].

This *lipophilicity code* was selected to convey information in an immediate and intuitive way: in fact, the white/smooth/reflective surface reminds us of materials such as wax or ceramic, that repel water; on the contrary, a darker, rough and opaque material is more reminiscent of brick or biscuits, which readily absorb water in real life.

With the steps described (calculation of motion that provides the position of atoms at each frame, calculation of surface, and texturing of the surface), it is possible to obtain a series of images that compose a movie showing the morphing of a protein between different conformations. For example, if the conformational change of a protein leads to exposure of patches of hydrophobic surfaces, this will become immediately evident in the movie. An example can be seen in our movie PROTEIN EXPRESSIONS—Study N.3 [20], at about min. 3.10.

11.4 Visualization of the Electrostatic Potential

If the shape and size of a protein and its lipophilicity are important features to consider in the study of molecular behavior, the electrostatic potential generated by its atomic components is at least as important. Few programs have been developed and distributed that can calculate such potential (e.g., [16, 21]). Its classical visualization is made using one of three techniques: color (red for negative and blue for positive, neutral white) on the surface, isopotential surfaces, or field lines.

The most widely used programs, such as VMD, Swiss-PDBviewer, or PyMOL, implement the “standard” color scheme described above. However, this scheme is not necessarily shared with other scientific communities, and it may cause confusion since in physics the “standard” is opposite: red for positive and blue for negative.

Of the programs above, VMD and spdbv even allow users to change colors, introducing a further level of possible ambiguity.

For this reason, we considered that a different system might be used, and we have deployed a code based on cinematic visual effects that describes field lines. The choice originated from the need to avoid obscuring the surface and its painted lipophilicity, of conveying the idea that EP has an effect at a distance, at least relative to lipophilicity (which instead is effective only very close to the surface), of being easy and intuitive for nonexpert viewers, and of being relatively quick to calculate and render.

Field lines are routinely used as a means for representing potential fields. In the case of electrostatic potential, lines are defined as the path that a hypothetical positively charged probe would run when placed in the field; therefore field lines naturally run toward the negative pole. Proteins and other biological molecules are composed of atoms, which may carry partial (or full) charges, and therefore create complex potential fields in their surrounding medium. These charges are often important in the determination of protein behavior in the presence of other (partially) charged molecules. Furthermore, as molecules move, both internally (as in changing conformation) and relative to each other, the charges associated with them also move, and correspondingly change the potential field.

Thus the visualization of EP in a moving scene is a nontrivial challenge, since it has to be calculated and represented at each step, and has to provide the viewer with consistent, unequivocal, and easy-to-understand information.

The implementation deployed with BioBlender, which entails calculation of the potential 5 times per second, and representation with a visual effect in which small linear particles travel along the field lines (from positive to negative) in about 1 s, aims at solving this challenge.

In the following, we describe the steps of the process of EP representation in BioBlender: calculation, data elaboration, and visualization.

Calculation of EP is based on APBS as shown in Fig. 11.2.

BioBlender first calls PDB2PQR [15], a program that reads a pdb file and associates a charge value to each atom, based on an internal library and on several inputs selected by the user and specified via the BioBlender interface. The output of PDB2PQR consists of two files: the PQR file, which is basically a modified version of pdb, and a file containing necessary information to be used by the next program, such as the dimension of the molecule, the ion concentration, and the dielectric constant of the medium. In the next step, APBS evaluates the electrostatic potential density in the space of the protein, by solving the Poisson–Boltzmann equation, in which the values of the partial charges associated with atoms are integrated, and the values of the potential are assigned to points in the grid. This program creates a file.dx which describes a box subdivided in a grid of the electrostatic potential values; the size of the grid is determined by the size of the protein, while its density can be defined by the user: a finer grid allows for better description, but is more time- and resource-consuming.

This grid of values (file.dx) is passed to scivis.exe, a program written in-house that converts it into a grid of vectors by trilinear interpolation at the center of each

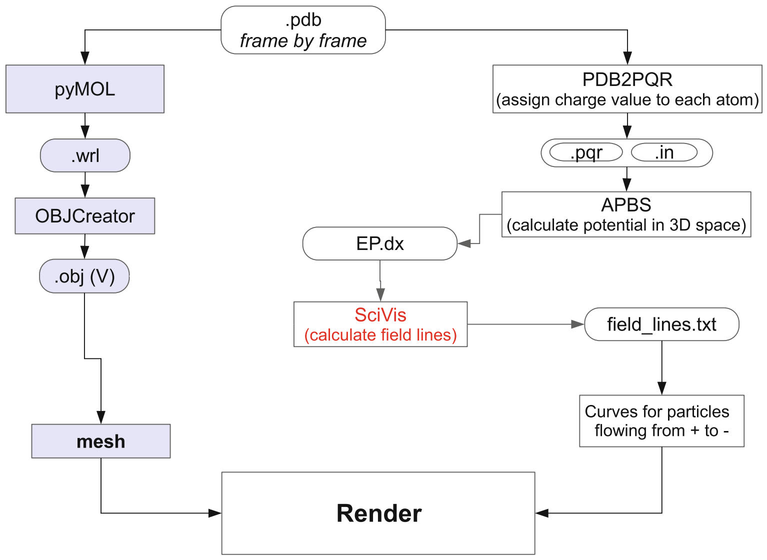


Fig. 11.2 Process for the definition of field lines in BioBlender

cube defined by the original grid. This passage makes it feasible to calculate lines by selecting only points of the vector grid, since in such a grid only one line can be drawn (in two directions) that passes through every point.

Selection of the points for drawing the lines is performed according to the following steps, summarized schematically in Fig. 11.3; a weighted Monte Carlo sampling procedure considers the points at the surface of the molecule, and the value of the vector at each point. This mechanism allows for the parts of the protein where the charges are exposed to be affected by a higher number of lines. Once the points are selected, lines are drawn following the gradient in both directions until one of three conditions is met: (i) the potential reaches zero (or a value determined by the user), (ii) the line enters the molecule, or (iii) the line reaches the end of the grid.

The total number of lines is proportional to the total charge of the molecule; this is to ensure that different proteins can be compared, so that a charged protein will have more lines than a neutral one. This is often not the case in other programs that allow users to decide how many lines to show, irrespective of the total charge.

Lines are encoded in .txt format and sent to Blender, which displays them with a visual effect composed of small white linear objects that run along the line from the more positive to the more negative end.

It is important to consider that during morphing the position of the local charges may change, and accordingly the potential must be recalculated and a new grid file generated. It can also happen that two opposite charges become close enough for them to effectively neutralize each other; in this case the system will show a bridge forming and apparently pulling the charges until they come into contact and disappear.

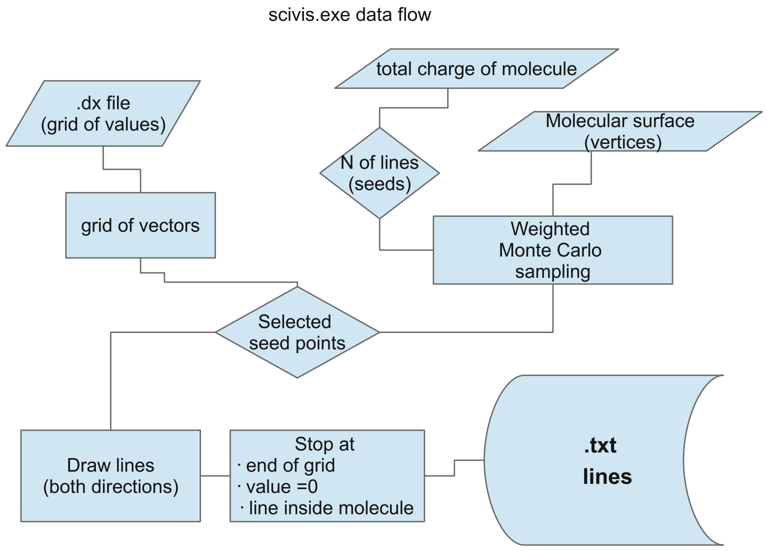


Fig. 11.3 Data flow in scivis.exe, for the selection and processing of field lines

For this reason, the EP is recalculated at short intervals (every 5 frames, i.e., 0.2 s), allowing for a smooth movement to be perceived by human eyes.

Display of the EP can be used in still images or animated movies; for the former, where the information provided by the direction of motion (positive to negative) cannot be included, the small lines are substituted by short comets, which are perceived as running "head on," i.e., with the head in the negative and the tail in the positive direction, as exemplified in Fig. 11.4.

An animated form of fruition is the interactive mode. In this case (see examples at http://www.scivis.it/3d-interactive/), the object (protein, group of molecules or other) is not moving, but the user can change the view by using the mouse. Lines are stored (together with other elements, such as mesh and texture) after having been calculated once, and the particle flow is continuous.

In the classic movie, prepared offline and observed on screen in a noninteractive form, the author of the clip can decide on the motion of the object and the camera in order to provide the desired information to the viewers.

Future expansion could explore the possibility of combining the interactive mode with moving objects. For example, a user might explore the possible interactions between two proteins, or more frequently, between a protein and a small molecule such as a drug. The protein can be presented in several conformations, or in the transition between them, allowing the user to test if and when in the transition the interaction can happen, and observing how the interaction evolves in time. The visualization system might be associated with sophisticated instruments for molecular dynamics, and/or with a visual environment such as an interactive cave to allow more direct forms of interaction.

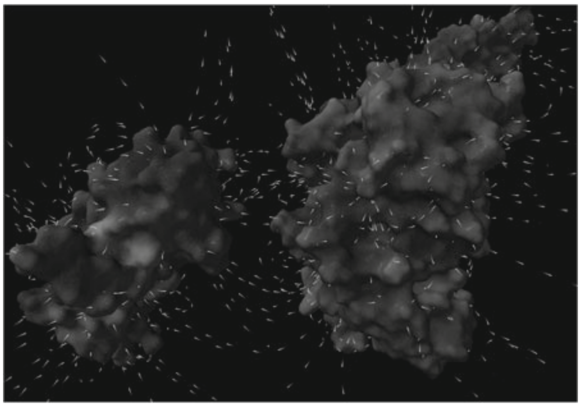


Fig. 11.4 EP representation in still image. The image shows Calmodulin and part of MLCK; a bridge is clearly forming between the two at the *center* of the image, while in the *upper* part two fields of the same polarity appear as repulsive

11.5 Conclusions

The increasing amount of information relative to molecules, their activities, and their interactions with other molecules and the cellular environment make it more and more important to be able to observe this "nanoworld" in a way that is both scientifically accurate and easily understood by humans. Our visual intelligence is still the best instrument we have to identify patterns from complex scenes, recognize recurrent forms and activities, and understand behaviors. This activity, which is initially guided by visual intuition, can subsequently be codified into rules and laws that step-by-step contribute to our general understanding of the complex biochemistry of life.

In order to activate this visual intelligence, we need tools that translate data into visual scenes in a way that is consistent and reproducible. The development of BioBlender is an attempt to provide such an instrument to the community of structural and cell biologists.

Acknowledgments We thank all members of the SciVis lab for generous help and discussion, and Alison Frank for English revision. We are also grateful to an anonymous reviewer who provided insightful and constructive comments.

References

1. Leonardo anatomical drawings, available online from The Metropolitan Museum of Arts, London. http://www.metmuseum.org/research/metpublications/Leonardo_da_Vinci_Anatomical_Drawings_from_the_Royal_Library_Windsor_Castle#
2. Vesalius A (1543) De humani corporis fabrica libri septem. http://vesalius.northwestern.edu/flash.html

3. Kendrew JC, Dickerson RE, Strandberg BE, Hart RG, Davies DR, Phillips DC, Shore VC (1960) Structure of myoglobin: a three-dimensional Fourier synthesis at 2 A resolution. Nature 185:422–427
4. Richards FM (1968) The matching of physical models to three-dimensional electrondensity maps: a simple optical device. J Mol Biol 37:225–230
5. Watson JD (1968) The double helix: a personal account of the discovery of the structure of DNA. Atheneum, New York
6. Humphrey W, Dalke A, Schulten K (1996) VMD: visual molecular dynamics. J Mol Graph 14:33–38
7. The PyMOL molecular graphics system, Version 1.5.0.4 Schrödinger, LLC. www.pymol.org
8. Swiss-PdbViewer. www.expasy.org/spdbv/
9. Molecular Maya. http://www.molecularmovies.com/toolkit/
10. Molecular flipbook. https://www.molecularflipbook.org
11. Johnson GT, Autin L, Goodsell DS, Sanner MF, Olson AJ (2011) ePMV embeds molecular modeling into professional animation software environments. Structure 19:293–303
12. Drew Berry's beautiful work. http://www.wehi.edu.au/education/wehitv/
13. Blender Foundation. www.blender.org
14. BioBlender by SciVis. www.bioblender.eu
15. Dolinsky TJ, Czodrowski P, Li H, Nielsen JE, Jensen JH, Klebe G, Baker NA (2007) PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. Nucl Acids Res 35:W522–W525. http://www.poissonboltzmann.org/pdb2pqr/
16. Baker NA, Sept D, Joseph S, Holst MJ, McCammon JA (2001) Electrostatics of nanosystems: application to microtubules and the ribosome. Proc Natl Acad Sci USA 98:10037–10041. http://www.poissonboltzmann.org/apbs/
17. Zini MF, Porozov Y, Andrei RM, Loni T, Caudai C, Zoppè M (2010) BioBlender: fast and efficient all atom morphing of proteins using blender game engine. arxiv.org/abs/1009.4801
18. http://bulletphysics.org/wordpress/
19. Andrei R, Callieri M, Zini MF, Loni T, Maraziti G, Pan MC, Zoppè M (2012) Intuitive visualization of surface properties of biomolecules. BMC Bioinform 13:S16
20. Zoppè M, Andrei RM, Cianchetta S, Loni T, Zini MF, Carlone I (2010) Video PROTEIN EXPRESSIONS—study N.3. https://vimeo.com/12363247
21. Rocchia W, Alexov E, Honig B (2001) Extending the applicability of the nonlinear Poisson-Boltzmann equation: multiple dielectric constants and multivalent ions. J Phys Chem B 105(28):6507–6514