Prof. Dr. Michael Krone
Marco Schäfer
Marcel Bok
Big Data Visual Analytics in Life Sciences
Department of Computer Science & IBMI
Faculty of Science

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Scientific Visualization

SS/2021

Assignment 7

Due: July 1st, 2021, 9:00 am

Code skeleton:

We supply you with a basic code template to use for this assignment but you are free to use the code you wrote for assignment one as a basis for this task. Our supplied code will contain code snippets assisting you in solving the tasks. So be sure to look at the supplied code if you choose to use your own code as a basis for this assignment.

1 Basic Molecular Visualization (16 pts)

For this Assignment the supplied code skeleton includes a PDB parser. PDB is data format to store atom positions and additional certain properties of molecules. (More information about the PDB format can be found here: `http://www.wwpdb.org/documentation/file-format-content/format33/sect9.html#ATOM`). Utilize the provided PDB parser to solve the following tasks.
Besides the code solution and the theoretical task also hand in screenshots for tasks 1.1, 1.3, 1.4 and 1.5 (Bonus) using `1crn.pdb`! We provided two PDBs, a smaller one containing a small molecule (`atp.pdb`) which is ideal for debugging but **does not contain meaningful temperature factors** and a larger one (`1crn.pdb`) containing a complete protein with meaningful temperature factors.

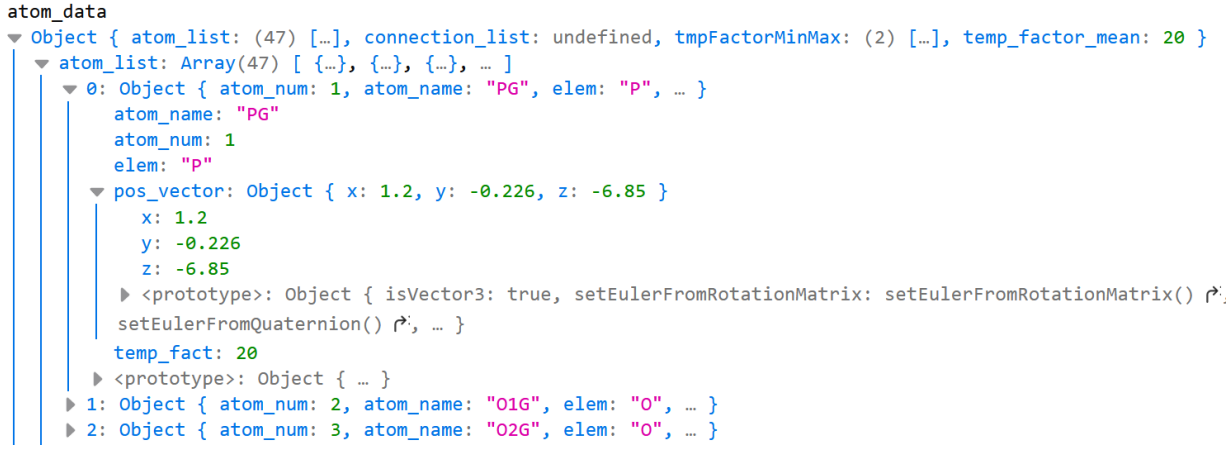

Figure 1: excerpt from data structure of parsed atom data

1.1 Mapping the Atoms to Colored Spheres (5 pts)

In molecular visualization, the "hard sphere" model is commonly used, where atoms are depicted as spheres (that is, the atoms are *mapped* to spheres).

Given the parsed PDB file, draw the atoms included in this file as spheres in the scene. Use the Van-der-Waals radii to define the radius of the spheres and the elements of the atoms as a basis for coloring (i.e., the element is *mapped* to a color). Keep in mind that the proper choice of geometry type can reduce the burden on your hardware. Both the Van-der-Waals radii and the Element-Color associations are supplied in the code skeleton inside the `constants.js` file as lookup tables. Place the function in `renderers.js`.

1.2 Calculate Connections (4 pts)

If the centers of two atoms are closer than 60 % of their added Van-der-Waals radii, it can be assumed that they form a covalent bond. Determine which atoms form a covalent bond for the given PDB. Note that performance/runtime is not important here, a brute-force calculation is fine! You should avoid duplicating bonds (e.g., only the connection between atom i to atom j, but not j to i). Template function `search_bonds()` in `utility.js` with function call in `pdb_parser.js`.

1.3 Draw Connections as lines (2 pts)

Draw the connections you calculated in the step before as lines between the atoms. Place the function in `renderers.js`.

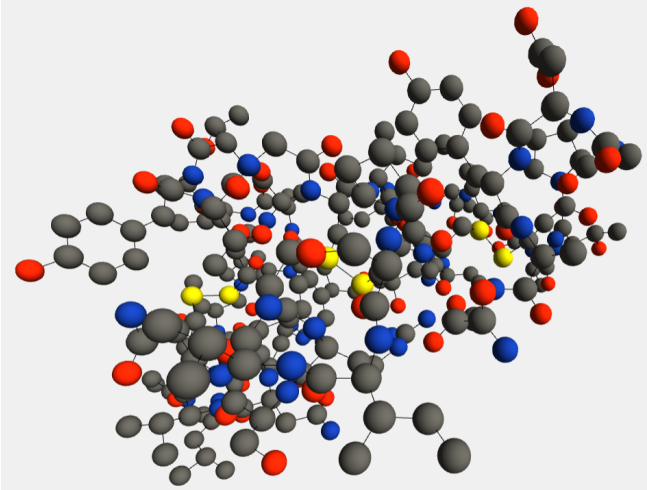

Figure 2: ball and line model of 1crn.pdb

1.4 Draw PDB as a Ball and Stick model (5 pts)

With the list of calculated connections, draw the connections as cylinders connecting the atoms. As the Van-der-Waals radius is too large for a proper Ball and Stick model, scale the atom radii by a factor of 0.25. Template function `calculate_connections_elements()` in `utility.js` should be used in a new render function, which is to be placed in `renderes.js`.

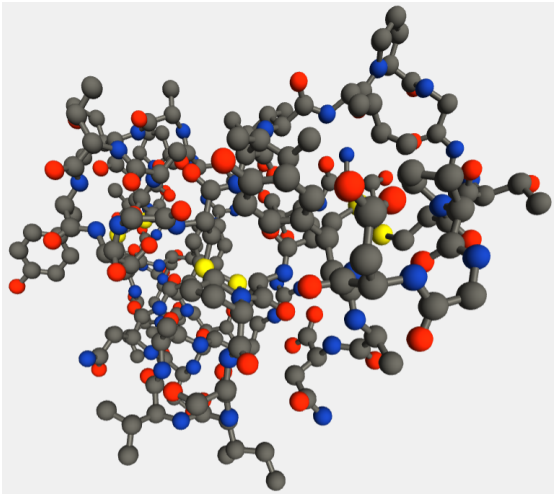

Figure 3: ball and stick model of 1crn.pdb

1.5 Bonus: B-Factor-based Coloring (2 pts)

We will now change the color mapping for the spheres so that it does not represent the element anymore. Instead, we will use another property, the co-called *B-factor*, which denotes the thermal vibration if the atom. To set the colors for each sphere, normalize the B-Factor values (stored in the variable `atom_data.temp_fact`) and map it to a diverging color scale from blue over white to red. Apply these colors to the spheres. Template function `tmpFactor_coloring()` in `utility.js`.

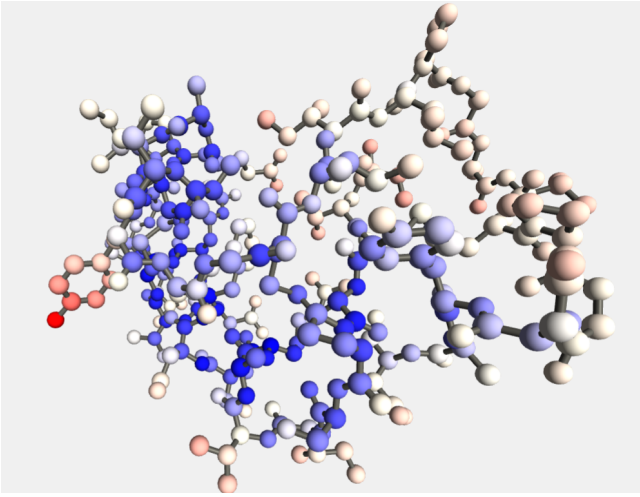

Figure 4: Bonus: B-Factor based coloring of 1crn.pdb

2 Theory

2.1 Color Interpolation on Triangle Grids (4 pts)

In the lecture and in the last exercise sheet, we looked at Barycentric interpolation for triangles and bilinear interpolation for quadrilaterals. Regarding the interpolation of colors, what problems can occur when having two adjacent triangles instead of a square? Briefly explain the advantages of having a triangulation that satisfies the Delaunay properties for (color) interpolation and make a drawing/sketch that supports your explanation.

Handing in

For the Hand-in, write the names of both group members at the top of all code files and PDF documents. Create a ZIP archive of your project folder and **PLEASE EXCLUDE** the folder **NODE_MODULES** and also the **main.js** file from your archive. Name it with the last and first name of each group member, and the assignment number (e.g., *schaefer_marco_bok_marcel_assignment7.zip*) and upload it to `Assignment_7`.