MASARYK
UNIVERSITY

FACULTY OF INFORMATICS

Modern visualization of partial atomic charges in Mol*

Bachelor's Thesis

DOMINIK TICHÝ

Brno, Spring 2023

MASARYK UNIVERSITY

FACULTY OF INFORMATICS

Modern visualization of partial atomic charges in Mol*

Bachelor's Thesis

DOMINIK TICHÝ

Advisor: RNDr. Tomáš Raček, Ph.D.

Department of Computer Systems and Communications

Brno, Spring 2023


FACULTAS ARTIS INFORMATICÆ
UNIVERSITAS MASARYKIANA

Declaration

I declare that I have worked on this thesis independently, using only the primary and secondary sources listed in the bibliography.

Dominik Tichý

Advisor: RNDr. Tomáš Raček, Ph.D.

Acknowledgements

TODO

Abstract

TODO

Keywords

Molstar, molecular visualization, partial atomic charges, molecular graphics, scientific visualization, web graphics, structural biology

Contents

List of Tables

List of Figures

Introduction

1 Theory

Theoretical concepts are fundamental to the study of computational chemistry, providing a framework for analyzing molecular structures and properties. This chapter focuses on four key areas of theory, beginning with an overview of molecular structure in Section 1.1. Section 1.2 provides an in-depth examination of chemical file formats, including the advantages and disadvantages of different file formats for storing molecular data. Partial atomic charges are explored in section 1.3, including their importance in the analysis of molecular structures and the various methods used to compute them. Finally, section 1.4 delves into color interpolation, a critical technique for visualizing partial atomic charges in molecular structures.

By providing a comprehensive overview of these theoretical concepts, this chapter aims to provide a strong foundation for the subsequent chapters, which will focus on the implementation and analysis of the Molstar extension for visualizing partial atomic charges in molecular structures.

1.1 Molecular structure

Molecular structure refers to the arrangement of atoms and chemical bonds in a molecule. The three main components of molecular structure are atoms, residues, and chains. In this section we will look at

1.1.1 Atoms

Atoms are the basic building blocks of matter. Atoms are composed of protons, neutrons, and electrons. The number of protons determines the element, while the number of neutrons determines the isotope. The number of electrons determines the charge of the atom. Atoms are the smallest unit of matter that can take part in a chemical reaction.

Bonds are the connections between atoms that hold molecules together. There are different types of bonds, e.g. covalent bonds, ionic bonds, and hydrogen bonds.

1.1.2 Residues

A residue refers to a specific building block that remains after a chemical modification or enzymatic reaction. In proteins, residues refer to amino acids, which are connected through peptide bonds to form polypeptide chains. Residues are crucial in biochemistry because they determine the structure and function of biological molecules. The sequence of residues in a protein or nucleic acid, for instance, determines its three-dimensional structure and ultimately its biological activity.

1.1.3 Chains

Polymer chains (chains) are sequences of residues that are linked together. In the context of biomolecules, chains can be either polypeptide chains in proteins or polynucleotide chains in nucleic acids. The sequence and structure of these chains are crucial for understanding the function and properties of the biomolecules.

1.2 Chemical file formats

TODO: introduction

1.2.1 SDF

Structure-data file (SDF) is a widely used chemical file format for representing molecular structures and their associated properties. It is a text-based format that describes the atoms, bonds, and atomic coordinates of a molecule.

1.2.2 MOL2

The Mol2 file format is another text-based format for storing molecular structures and their associated properties. It can store multiple conformations of a molecule and is commonly used in molecular modeling and cheminformatics applications. The Mol2 format provides more flexibility and additional features compared to the SDF format, such as support for multiple substructures and atom types.

1.2.3 PDB

The Protein Data Bank (PDB) file format is a widely used format for storing three-dimensional structures of proteins, nucleic acids, and other macromolecules. PDB files contain information about the atomic coordinates, secondary structure, and other important details required for understanding macromolecular structures. The PDB format has been widely adopted in structural biology, bioinformatics, and related fields.

1.2.4 mmCIF

The macromolecular Crystallographic Information File (mmCIF) format is an extension of the CIF format, specifically designed for macromolecular structures. It is a text-based format that provides a more comprehensive and flexible representation of macromolecular crystallography data compared to the PDB format. One of the most important features of the mmCIF format is its support for data dictionaries. This allows users to define new data items and integrate additional information. In contrast to other formats, the mmCIF format does not impose limits on column width and entry count, making it more flexible and accommodating for storing large amounts of data.

TODO: add example image + better explanation

1.3 Partial atomic charges

Partial atomic charges are a measure of the distribution of electronic charge within a molecule. These charges are important for understanding and predicting molecular interactions, including hydrogen bonding, electrostatic interactions, and solvation effects.

TODO: visual of electron distribution

1.3.1 Calculation methods

1.3.2 ChargeFW2

(1)

1.3.3 AlphaCharges

1.4 Color interpolation

Color interpolation is the process of creating new colors by mixing two or more colors together. It is a common technique used in computer graphics and digital image processing to create smooth transitions between colors.

Color interpolation works by calculating the intermediate colors between two or more given colors. This is typically done by taking a weighted average of the red, green, and blue values of the colors being interpolated.

TODO: add math equation + image of red,white and white,blue interpolation

1.5 Provider pattern

2 Visualizing molecular data

This chapter will discuss various types of visualizations, the role of coloring in molecular representations, and some commonly used software tools for creating these visualizations.

2.1 Types of visualizations

There are several methods to represent molecular data, each with its own benefits and drawbacks. The methods most relevant to this work are the following three types: ball and stick, surface, and cartoon.

2.1.1 Ball and stick

The ball and stick model represents atoms as spheres and bonds as cylindrical connections between these spheres. This model provides a simple and intuitive visualization of a molecule's atomic structure. It highlights individual atoms and their bonds, including their bond types. However, it may not accurately represent the spatial relationships between atoms in larger molecules or macromolecular complexes.

2.1.2 Surface

Surface representations depict the three-dimensional shape of a molecule by displaying its solvent-accessible surface. This model provides a more accurate representation of the molecule's overall shape and size, making it especially useful for studying macromolecular interactions and the binding of small molecules. For example, surface visualization can be used to identify potential binding sites on a protein surface, which can then be targeted by drug molecules.

2.1.3 Cartoon

Cartoon representations simplify the molecular structure by focusing on the secondary structure elements of proteins and nucleic acids, such as alpha helices, beta sheets, and loops. Alpha helices are often

depicted as a spiral-like structures, whereas beta sheets as arrows. This type of visualization is particularly useful for visualizing large macromolecular complexes, as it highlights the overall organization and topology of the molecule without the clutter of atomic details. The simplification of the structure also makes it easier to understand the folding and dynamics of the molecule.

2.1.4 Coloring of molecular visualizations

Coloring is an essential aspect of molecular visualization, as it can provide additional information and help to emphasize specific features or properties of the molecule. Some common coloring schemes include: - By element: Atoms are colored according to their chemical element (e.g., carbon in grey, oxygen in red, nitrogen in blue). - By partial atomic charge: Atoms or residues are colored according to their charge or charge sum. Negative charges are depicted in red, positive charges in blue.

2.2 Visualization software

There are numerous software tools available for visualizing molecular data, with varying levels of complexity, customization, and features. Two widely used tools are LiteMol and Mol*.

2.2.1 Litemol

LiteMol is an open-source, web-native molecular visualization tool that supports various file formats and offers a user-friendly interface for creating visualizations. LiteMol provides essential visualization types, including ball and stick, surface, and cartoon representations, as well as options for customizing colors, lighting, and other display settings. The web-based nature of LiteMol makes it easily accessible and platform-independent.

The LiteMol suite is a freely available tool for visualizing large macromolecular structure datasets, which consists of three components: data delivery services, a compression format, and a lightweight 3D molecular viewer. It enables fast delivery and visualization of large datasets and is compatible with modern web browsers and mobile

devices, making it accessible to users with and without structural biology expertise. The tool addresses the challenges of delivering and visualizing large structural data sets, which are becoming increasingly available due to advances in electron microscopy and other techniques. (2)

2.2.2 Molstar

(3)

Mol* (Molstar) is another web-native molecular visualization tool, developed as part of the wwPDB OneDep system for macromolecular structure deposition and validation. Mol* offers a wide range of visualization options, including advanced features such as electron density maps and validation reports. Mol* supports many file formats, including PDB, mmCIF, and PDBx/mmJSON. Like LiteMol, Mol* is platform-independent and can be accessed from any web browser.

Mol* emphasizes interactivity and offers various tools for manipulating and analyzing the molecular structure, such as distance and angle measurements, selection and display of specific residues, and custom coloring schemes. Additionally, Mol* provides integration with external databases and services, such as UniProt, PDBe, and RCSB PDB, enabling users to quickly access related information and resources.

3 Molstar partial charges extension

Visualizing partial atomic charges in molecules is an essential aspect of computational chemistry research, aiding in analyzing complex molecular structures. Molstar provides an extensive range of features for users to explore molecular structures. However, the tool lacks the functionality to color and label atoms and residues based on their partial atomic charges. This can be a considerable limitation for researchers. In response to this need, we have created an extension to Molstar that addresses this limitation.

This chapter describes the requirements for the extension, the custom mmCIF categories necessary for storing the partial atomic charges, and the implementation of the extension itself.

3.1 Requirements

This section briefly describes the formal requirements for the Molstar extension. Firstly, the extension should enable the coloring of atoms and residues based on their partial atomic charges. Secondly, it should describe the charge values of the atoms and residues. Thirdly, the extension should allow the user to provide multiple charge sets for a single structure and select which one to display. Finally, the extension should be seamlessly integrated into the Molstar library, facilitating access to its features and functionality.

3.2 Custom mmCIF categories

To store partial atomic charges within a single file, we developed custom categories within the mmCIF format. The mmCIF format was chosen because it is widely used in the field of structural biology and offers several advantages over other formats, as discussed in 1.2.4. The custom categories allow us to store information about the partial atomic charges separately from the other structural data, while still being able to access it within the same file. Storing all data in one file was important as it allowed for easier management and distribution

of the data. If the charges were stored separately, we would have had to create custom import controls in Molstar to bring them in.

We used two separate categories for this purpose: one to store the partial charge values for each atom in the structure, and another to store metadata about the charge sets.

We chose to use the mmCIF file format because it offered several advantages, as previously discussed in Section 1.2. The most significant advantages for our purposes was the ease of creating custom data categories to describe data related to the partial atomic charges and therefore the ability to store all the data in a single file. And since the mmCIF format is widely used in the field of structural biology, it was a natural choice for our extension.

The custom categories allowed us to store information about the partial atomic charges separately from the other structural data. We used two separate categories for this purpose: one to store the partial charge values for each atom in the structure, and another to store metadata about the charge sets.

The first category binds the atom and its charge using an attribute called atom_id, which points to the atom_site.id item. Additionally, we used an attribute called typeId to point into the second category, which is dedicated to storing metadata about the charge sets.

The metadata category has three attributes: id, type, and method. The id attribute is a unique identifier for the charge set, the type attribute describes the type of computation method used (e.g., empirical, quantum, etc.), and the method attribute describes the method name with parameters if applicable (e.g., 'EEM/Racek 2016 (ccd2016_npa)').

Figure 3.1 provides a detailed illustration of the structure of the custom mmCIF categories.

TODO: rewrite

However, in order to store partial atomic charges within a single file, we needed to create custom categories within the mmCIF format. The design of these custom categories was a crucial step in creating the extension since we needed a way to package the charge data together with the structure data into one file.

Without having everything in one file we would have to provide the charges in a different way e.g. through custom import controls.

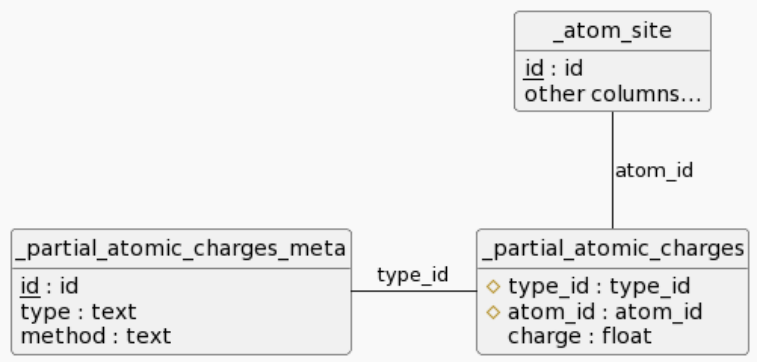

Figure 3.1: Diagram of custom mmCIF categories

For this purpose we have chosen the mmCIF file format. There are a couple of reasons already discussed in section 1.2 as to why mmCIF is more advance file format than the other formats discussed in section 1.2. For our purposes the biggest advantage was the ease of creating custom data categories for describing data relating to the partial atomic charges.

The custom categories allowed us to store information about the partial atomic charges separately from the other structural data.

We used two separate categories for storing the charge data. The first category stores the partial charge values for each atom in the structure. The binding of atom and its charge is done using an attribute atom_id which is a pointer to the atom_site.id. item. Additionally, we use an attribute typeId as a pointer into the second category which is dedicated to storing metadata about the charge sets.

The metadata category has three attributes - id, type, and method. Attribute id is a unique identifier for the charge set, attribute type serves to describe the type of computation method used (e.g. empirical, quantum etc.), and lastly the method attribute describes the method name with parameters if applicable (e.g. 'EEM/Racek 2016 (ccd2016_npa)').

Figure 3.1 provides a detailed illustration of the structure of the custom mmCIF categories.

```typescript
type TypeId = number;
type IdToCharge = Map<number, number>;

export interface SBNcbrPartialChargeData {
    typeIdToMethod: Map<TypeId, string>;
    typeIdToAtomIdToCharge: Map<TypeId, IdToCharge>;
    typeIdToResidueToCharge: Map<TypeId, IdToCharge>;
    maxAbsoluteAtomCharges: IdToCharge;
    maxAbsoluteResidueCharges: IdToCharge;
    maxAbsoluteAtomChargeAll: number;
}
```

Figure 3.2: Interface of the custom model property used for storing the partial atomic charges.

3.3 Implementation

This section will detail the implementation of the extension. The extension consists of multiple providers that are registered in a global registry. Each provider serves a distinct purpose, which will be described in the following subsections.

The extension was created using TypeScript, a superset of JavaScript that adds static typing and other features to the language. The Molstar library is also written in TypeScript, so the extension was written in the same language to ensure compatibility.

3.3.1 Charges provider

TODO: rewrite this

This provider is responsible for supplying the partial atomic charges to the rest of the extension providers. In order to retrieve the charges from the mmCIF file, it is necessary to parse the file. This is already done by the Molstar library, which parses the mmCIF file and provides the parsed mmCIF file data in the form of a `MmcifFormat` object. This

object is then used to retrieve the custom mmCIF categories described in Section 3.2.

After retrieving the custom categories, the provider creates a custom model property for storing the charges. The interface depicted in Figure 3.2 describes the structure of the custom model property.

The atom charges are stored in the `typeIdToAtomIdToCharge` map. The map is indexed by the charge set (typeId) and the atom id. The atom id is a pointer to the atom_site.id. item in the mmCIF file. The atom charges are retrieved from the mmCIF file by iterating over the atom_site.id category and retrieving the charge values for each atom. The charge values are then stored in the `typeIdToAtomIdToCharge` map.

The residue charges are calculated by summing the charges of the atoms that make up the residue. This is done by iterating over the atoms of the residue and summing their charges. The residue charge is then stored in the `typeIdToResidueIdToCharge` map.

The maximum absolute charge values of the atoms and residues are calculated and stored in the `maxAbsoluteAtomCharges` and `maxAbsoluteResidueCharges` maps. These maps are used in the color theme provider to normalize the charges to the range of 0 to 1. Additionally, the maximum absolute charge values are used to calculate the color interpolations in the color theme provider. Additionally, the maximum absolute charge of both atoms and residues is calculated and stored in the `maxAbsoluteChargesAll` map.

Lastly, the method name used to calculate the charges of a given charge set is stored in the `typeIdToMethod` map. This map is used to display the method name in the UIs.

3.3.2 Color theme provider

This provider serves as the central component of the extension, with its primary function being to assign colors to atoms and residues based on their charges. It achieves this by using the ColorTheme API provided by Molstar. The ColorTheme API is a mechanism for assigning colors to structural elements of a molecule. These structural elements can be atoms, residues, bonds, and so on. The API is based on the concept of a ColorTheme object, which is a collection of color assignments

for structural elements. The ColorTheme object is then used by the Molstar library to color the structural elements of the molecule.

For the purposes of this extension, it was necessary to color two structural elements - atoms and residues. For both of these structural elements the charges were retrieved from the provider described in the previous section 3.3.1, which provided charges for atoms and residues.

To establish the color for a given charge, two color interpolations are employed: one for negative charges and another for positive charges. Atoms with positive charges receive a color from a white-to-blue color interpolation, while atoms with negative charges are assigned a color from a white-to-red color interpolation. These color interpolations are highlighted in Figure **??**.

3.3.3 Labels

Having colored the structural elements, it was also necessary to create a label provider, which would assign labels that describe the charge of the structural element. In order to determine which element is highlighted, Molstar uses the object Loci. A Loci object is utilized for general selections and highlights. Consequently, it is essential to first extract the location from the Loci object in order to obtain the atom ID. The charge is acquired from the property provider, and the label is an HTML string that conveys the charge of the atom or residue. An example of the label can be seen in in the right-hand corner in figure **??**.

3.4 Molstar integration

After creating the extension, it was integrated into the Molstar library. The extension is integrated in a way that allows the user to simply upload a mmCIF file together with the custom charge categories in the Molstar viewer. The color theme is set automatically to the partial atomic charge color theme. By integrating it into the Molstar library the extension was made freely available to anyone to use

3.5 Future work

The extension can be improved in many areas - mainly in the UI. It can be optimized by storing the atom and residue charge maps statically. The extension unnecessarily recalculates the maps when switching to another charge set. Another area that could use some work is the UI. When a user wants to switch to another charge set, they need to use the State Tree UI. If the user has only one structure loaded, this is fine. However, once you have more than one structure and want to change the charge set for each one of them, then it becomes very tedious. A better approach is to create a custom UI element in the right sidebar which would serve as a shortcut to set a charge set globally or for each structure individually. Another custom UI control could be used for setting a global max absolute charge - this is also done manually for each structure.

4 Molstar viewer plugin

TODO: maybe make this a section in chapter about ACC2

In addition to creating the partial atomic charges extension for Molstar, it was necessary to create a custom Molstar viewer instance to facilitate custom functionality not present in the official Molstar viewer instance deployed at molstar.org/viewer. This chapter focuses on describing the implementation and functionality of the custom Molstar viewer instance for the web applications discussed in chapters 5 and 6.

4.1 Requirements

The viewer needs extended functionality

5 Atomic Charge Calculator II

TODO: fix weak link between description of the application and the limitations

Atomic Charge Calculator II (ACC2) is a web application that calculates partial atomic charges for input structure files. The application is built using Flask for the backend and Javascript with Bootstrap for the frontend. It uses ChargeFW2 to perform the charge calculations and the Litemol viewer to visualize the structures with partial atomic charges. (1) However, the Litemol viewer has some limitations: it cannot handle multiple charge sets, and more importantly, it is no longer supported. Therefore, it was necessary to update the application to use the Molstar viewer, which enables multiple charge sets through the partial atomic charges extension described in Chapter 3.

This chapter describes the changes made to the ACC2 application to integrate the Molstar viewer. We first discuss the modifications to the ChargeFW2 output, then explain the changes made to the Flask backend to support multiple charges, and finally describe the frontend updates required to generate multiple charge set calculations.

5.1 ChargeFW2 extension

As mentioned in 1.3.2, ChargeFW2 is a C++ application for calculating partial atomic charges. To accommodate the specified output file format discussed in section, ChargeFW2 required an extension that would output a single mmCIF file containing both the molecular structure and charges. The charges were appended to the end of the file.jop

5.2 Multicharge support

5.2.1 Backend

5.2.2 Frontend

5.3 Molstar viewer integration

6 AlphaCharges

6.1 Viewer extension

6.2 Molstar viewer integration

Conclusion

Things were created and some things even work.

Bibliography

1. RAČEK, Tomáš; SCHINDLER, Ondřej; TOUŠEK, Dominik; HORSKÝ, Vladimír; BERKA, Karel; KOČA, Jaroslav; SVOBODOVÁ, Radka. Atomic Charge Calculator II: web-based tool for the calculation of partial atomic charges. *Nucleic Acids Research*. 2020, vol. 48, no. W1, W591–W596. ISSN 0305-1048. Available from DOI: 10.1093/nar/gkaa367.
2. SEHNAL, David; DESHPANDE, Mandar; VAŘEKOVÁ, Radka Svobodová; MIR, Saqib; BERKA, Karel; MIDLIK, Adam; PRAVDA, Lukáš; VELANKAR, Sameer; KOČA, Jaroslav. LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data. *Nature Methods*. 2017, vol. 14, no. 12, pp. 1121–1122. ISSN 1548-7105. Available from DOI: 10.1038/nmeth.4499.
3. SEHNAL, David; BITTRICH, Sebastian; DESHPANDE, Mandar; SVOBODOVÁ, Radka; BERKA, Karel; BAZGIER, Václav; VELANKAR, Sameer; BURLEY, Stephen K; KOČA, Jaroslav; ROSE, Alexander S. Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic Acids Research*. 2021, vol. 49, no. W1, W431–W437. ISSN 0305-1048. Available from DOI: 10.1093/nar/gkab314.