

Visual Evaluation of Computational Models of the Biological Mesoscale

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Elektrotechnik und Informationstechnik

eingereicht von

Guillermo García-Escribano,

Matrikelnummer 01635324

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Tobias Klein, M.Sc.

Mitwirkung: Peter Mindek, Dr.techn.

Wien, 20. Juni 2017

Guillermo García-Escribano

Tobias Klein

Visual Evaluation of Computational Models of the Biological Mesoscale

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Electrical Engineering and Information Technology

by

Guillermo García-Escribano,

Registration Number 01635324

to the Faculty of Informatics

at the TU Wien

Advisor: Tobias Klein, M.Sc.

Assistance: Peter Mindek, Dr.techn.

Vienna, 20th June, 2017

Guillermo García-Escribano

Tobias Klein

Erklärung zur Verfassung der Arbeit

Guillermo García-Escribano,
Molkereistraße 1, 1020 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 20. Juni 2017

Guillermo García-Escribano

Danksagung

Jetzt, da meine Tage der Arbeit an meiner Bachelor-Arbeit zu Ende gehen, ist es höchste Zeit, dass ich ein paar Zeilen schrieb, um meine Dankbarkeit gegenüber jedem auszu-drücken, der dies möglich gemacht hat. Zum Dank verpflichtet bin ich meinen Tutoren Tobias Klein und Peter Mindek, die mutig genug waren, die Aufgabe zu übernehmen mich durch das gewagte Abenteuer zu begleiten, das begann, als ich beschloss, ein Projekt auf dem Gebiet der biologischen Datenvisualisierung aufzunehmen nur um mehr über die Welt um uns herum zu lernen. Ihre Geduld und Unterstützung war ein Grundpfeiler für den Erfolg dieser Arbeit. Außerdem möchte ich mich herzlich bei meiner Familie bedanken, welche mich während des gesamten Projekts ermutigt und unterstützt hat, und bei meinen Freunde, welche meinen Aufenthalt in Wien zu einem herausragenden, einmaligen Erlebnis gemacht haben. Diese Arbeit widme ich allen erwähnten Personen.

Acknowledgements

Now that my days of working on my bachelor's thesis are coming to an end, it is high time I wrote a couple of lines to express my gratitude towards everyone that made this possible. I am indebted to my tutors Tobias Klein and Peter Mindek, who were brave enough to accept the task of guiding me through the daring adventure that began when I decided to take up a project in the field of biological data visualization just for the sake of learning more about the world around us. Their patience and support was a fundamental pillar for the success of this thesis.

Besides, I want to sincerely thank my family for encouraging and supporting me through this process, and my friends for making my stay in Vienna an outstanding once-in-a-lifetime experience. This thesis is dedicated to all of them.

Kurzfassung

Die derzeit verfügbaren Techniken zur Erfassung von Makromolekülen auf atomarem Niveau sind für große Strukturen der biologischen Mesoskala nicht geeignet. Daher müssen diese Strukturen wie Viren oder Zellorganellen aus molekularen Bausteinen mit Hilfe von Software-Tools zusammengestellt werden. Das Ziel der jüngsten Projekte wie cellPACK ist es, Modelle mit diesen Werkzeugen zu erstellen, so dass die wissenschaftliche Gemeinschaft iterativ Feedback geben und die Modelle bearbeiten kann, um schließlich die am besten geeignete Darstellung mit dem aktuellen Wissensstand zu erzeugen. Zu diesem Zweck müssen wir die Werte für Eigenschaften wie Verteilung, Dichte oder Opazität erkennen, die ein Modell wünschenswert machen. Diese Arbeit zielt darauf ab, ein Softwareprogramm zur visuellen Bewertung der Qualität der zusammengesetzten Strukturen zu schaffen. Das Programm extrahiert die Informationen über die Qualität der räumlichen Verteilung der Moleküle in den Szenen, die durch packing Algorithmen produziert werden, und zeichnen sie in einen Satz von 2D-Darstellungen. Diese vermitteln die statistischen Informationen über die Verteilung und ermöglichen den visuellen Vergleich von generierten Modellen, die sich nicht nur aufgrund der stochastischen Natur der packing Algorithmen, sondern auch wegen der Verwendung unterschiedlicher Parametereinstellungen ändern.

Abstract

Currently available techniques for capturing macromolecules on atomic level are not appropriate for large structures on the biological mesoscale. Therefore, those structures, such as viruses or cell organelles, have to be assembled from molecular building blocks using software tools. The goal of recent projects like cellPACK is to create models with these tools, allowing the scientific community to iteratively give feedback and edit the models, in order to eventually generate the most suitable illustration consistent with the current state of knowledge. For that purpose, we need to discern the values for properties like distribution, density or opacity that make a model preferable to others. This thesis aims to create a software program for visual evaluation of the quality of the assembled structures. The program will extract the information about the quality of spatial distribution of molecules in the scenes produced by packing tools and plot it into a set of 2D representations. These will convey the statistical information about the distribution and enable the visual comparison of generated models, which vary not only due to the stochastic nature of the packing algorithms but also because of the use of different parameter settings.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
1 Introduction	1
2 Related work	3
2.1 cellPACK	3
2.2 Comparative ensemble visualization	4
3 Methodology	7
3.1 Overview	7
3.2 Analysis tools	8
4 Implementation	13
4.1 Pilot program	13
4.2 Evaluation of cellPACK models	13
5 Results	17
5.1 Two-by-two model contrast	19
5.2 Potential and limitations	19
6 Conclusion and outlook	23
Bibliography	25

Introduction

Visualization of biological data is a key technique for researchers, students and educators to understand, communicate and learn from these data while gaining insight into the behaviour and structure of biological systems. Large structures like cells or bacteria can be easily observed by light microscopy. However, the resolution of light microscopy is limited by light diffraction to an approximate value of 100nm, being new techniques with better resolution incompatible with live imaging of living organisms [1]. On the other hand, the atomic structure of a given macromolecule like proteins or phospholipids can be determined by nuclear magnetic resonance spectroscopy or x-ray crystallography [2]. Nevertheless, there is currently no optimal method to obtain an atomic detailed view of those structures in the biological mesoscale (10-100nm), since they are too small for microscopy but too large and complex for the atomic detailed resolution techniques [14]. This range of resolution includes biological organisms such as cell organelles, antibodies or viruses [4]. Studying their characteristics and behaviour is a fundamental requirement for the understanding of biological systems. Since no direct observation is possible, the visualization of the mesoscale relies on different sources of data to create a computational model, which is a complex nonlinear system containing numerous parameters and related variables [5], including ultrastructures, subsystems and the connection between the different objects that compose the system. Given that there is no trivial solution for a computational model, the system is used to run simulations and study the possible outcomes. The source data for biological mesoscale models include information about atomic structures, concentrations, molecular interactions or constraints, and are the basis for the generation of a specific packing process for the construction of an illustrative composition [6].

Creating a virtual model of the mesoscale requires solving a biological nontrivial packing problem, because the main difficulty lies in placing the biological subsystems inside a container with limited space, like a cell membrane, while respecting the knowledge about their true behaviour and appearance. Unlike other simpler optimization problems where

the only goal is to pack geometric objects as dense as possible[7], biological packing problems implies dealing with diverse sizes, shapes, properties, location frequencies and interactions between objects [8]. This is cumbersome and computationally expensive. To generate those views in an automatic and easy manner we can use packing algorithms like cellPACK[9], which is an interactive software toolkit that creates mesoscale models from the available data concerning the composition of the structure, which they refer to as the ‘recipe’ of the model. This is done by placing ‘ingredients’ into containers while respecting the natural randomness of biology and the interactions between the different ingredients. cellPack provides us with the tools to interact with the models and modify the ingredients and recipes stored in a public online database, generating different results for the same visualization.

If we take into consideration the numerous and diverse parameters that influence the outcome, it is natural to wonder which factors have a greater repercussion in creating an optimal model. For instance, an algorithm that prioritizes the packing of bigger 2D objects as densely as possible will derive in a heterogeneous model with a larger number of objects in the outer parts of the picture, while a random placement of the ingredients will lead to a much more homogeneous result [8]. The same happens when we change the value of the parameters that form the recipe. If for example we set the concentration of the bigger objects to be greater around the centre of the picture, or the location frequency of the smaller ones to be larger on the periphery, we will obtain again a heterogeneous illustration. However, changing other parameters could have an insignificant influence on the output. For this reason, it is necessary to have a tool to analyse and evaluate the varying results of a visualization model, to study both the effects of probabilistic packing and the influence of different parameter settings. Once we understand how slight changes in a recipe affect the created structure and how a recipe can have as a result a broad range of different models, all of which are compatible with the current state of knowledge, we will gain a better insight into the behaviour of living organisms as we improve the visualization of the biological mesoscale.

The aim of this thesis is to create a software program for visual evaluation of the quality of the assembled mesoscale structures generated by automatic packing algorithms, with the intention of laying the foundations for a tool that will enable researchers to easily find aspects of the available modelling software that should be improved.

Related work

Visualization of the biological mesoscale is a young and booming field of research. Whereas there was no method to observe the mesoscale in atomic-resolution detail back in 2014[8], nowadays there is a range of models of mesoscale structures compiled in an open source online database [9]. Currently these models are still being iteratively improved while new ones are being created with the data provided by the scientific community, with the goal of being able to obtain a complete, dynamic and accurate view of the biological mesoscale at some point in the future.

2.1 cellPACK

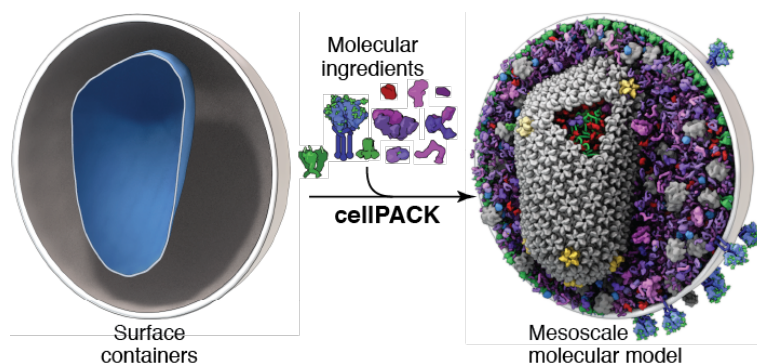


Figure 2.1: cellPACK creates 3D models of the biological mesoscale by packing from a recipe of molecular ingredients into a set of surface containers.

Originally, illustrations of biological structures such as the HIV virion were drawn manually. Even if the artistic approach respected the same recipe that software tools use to produce 3D models, the outcome was limited to 2D paintings with low or non-existent

atomic resolution, as described by Johnson et al.[6] This method was inconvenient for the length of the process and the lack of interactivity of the final result. For this reason, the appearance of modelling software toolkits like cellPACK was a huge step towards modern mesoscale visualization. When given a dataset, cellPACK packs the desired containers with ingredients according to that recipe, as displayed in Figure 2.1. As a result, the program creates a 3D model of a biological structure that can be explored and analysed. Since the distribution is stochastic, there are a number of possible models that can result from the recipe. One of the advantages of cellPACK is that it lets users actively explore and simulate recipes, allowing for subsequent data edition and model updates. The main issue with cellPACK is the fact that creating a new composition from a recipe can take from minutes to hours and so the software relies greatly in already implemented models. This has been solved recently by new approaches that make it possible to generate new models in a matter of seconds by applying parallel processing [10]. Thanks to the large number of models that this method makes available, not only the possibility, but also the need to analyse ensemble of models arises. In particular, it is interesting to study both the difference between models caused by the alteration of the recipe for a same biological structure and the difference caused by the stochastic packing of the ingredients for a same recipe.

2.2 Comparative ensemble visualization

Different methods have been proposed to analyse a collection of numerous datasets representing divergent compositions. Most approaches involve a visual exploration of the models rather than the use of raw quantitative data. Since the aim of this thesis is to provide a tool for the simultaneous visual evaluation of different mesoscale models, we will base our tool on the available literature on ensemble visualization. By comparing and contrasting the models generated by packing software like cellPACK, we will be able to reach conclusions concerning the quality of the compositions. In the following sections we will take a look into different approaches to comparative ensemble visualization proposed by various authors.

2.2.1 Parameter space exploration of ensemble datasets

Phadke et al. [11] underlined the need to identify shapes, patterns, distributions and unusual values in each model as well as the need to compare those factors simultaneously across multiple members. Their work describes two approaches for this matter: ‘pairwise sequential animation’ where a model is created in which values and members can be distinguished by the size, opacity, shape and colour of the ingredients, and ‘screen door tinting’ where similarities and differences between models are highlighted by the use of saturation tinting. An example of the first method can be seen in Figure 2.2a. Another technique to visualize and compare ensemble datasets is the so-called ‘multi-image view’[12], as displayed in Figure 2.2b. The 2D illustration of the model is built by tightly packed hexagons (base tiles) which are divided into sections. In each of these sections

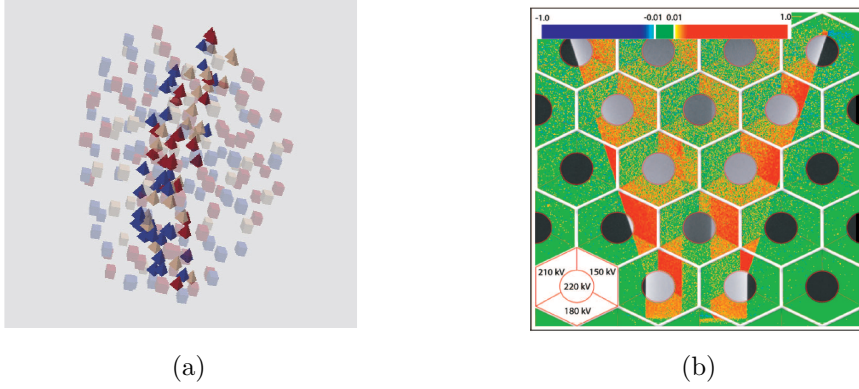


Figure 2.2: In (a) we see an example of pairwise sequential animation, with a banana being represented by pyramids and an apple with cubes. The colours stand for different properties of the objects. On the right, subfigure (b), there is a multi-image view of a 3D X-ray computed tomography where the central dataset is seen as a gray value and the difference of the other datasets to the central one are colour coded[11][12].

we display a different dataset with one of them acting as the central dataset, which neighbours every other one. The deviations between the models are colour-coded and displayed on screen in order for the user to have a visual impression of the discrepancy between them.

2.2.2 Statistics as a source of visual information

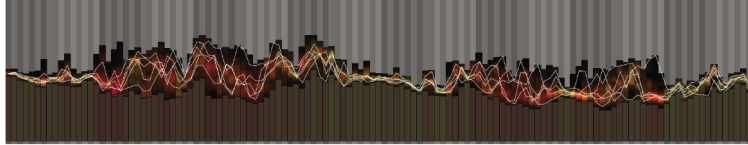


Figure 2.3: Multi-chart showing statistical quantities encoded in the histogram and single ensemble member values displayed via polylines.[13]

Moreover, some approaches to the analysis of computational models include the use of comparative charts for volumetric data. A good example for this is the work of Demir et al.[13] who propose multi-chart visualization for ensemble data. An example is shown in Figure 2.3 In this interactive technique, bar charts showcase the statistical information concerning ingredient distribution and density, while an overlaid line chart displays the variation between ensemble members. Whereas this tool seems more complex and abstract than the previously mentioned techniques, the authors emphasise that users spent more time in this visualization mode than in the 3D view of the model that is easily accessible from the chart.

2.2.3 Global approach to visual evaluation of the mesoscale

Finally, it is essential to mention that the creators of cellPACK have proposed to use along with their modelling software a set of basic tools to analyse the results of their algorithms[8]. These tools include both aforementioned approaches: a visual display that highlights empty spaces and edge effects, and a histogram that evaluates the uniformity of the ingredient distribution in a 2D illustration. They are both more basic approaches than the previously mentioned techniques. Since the expectation of this thesis is to build a software tool to give the most general and relevant information about the evaluated models, the program will be designed according to this idea of a global analysis.

Methodology

3.1 Overview

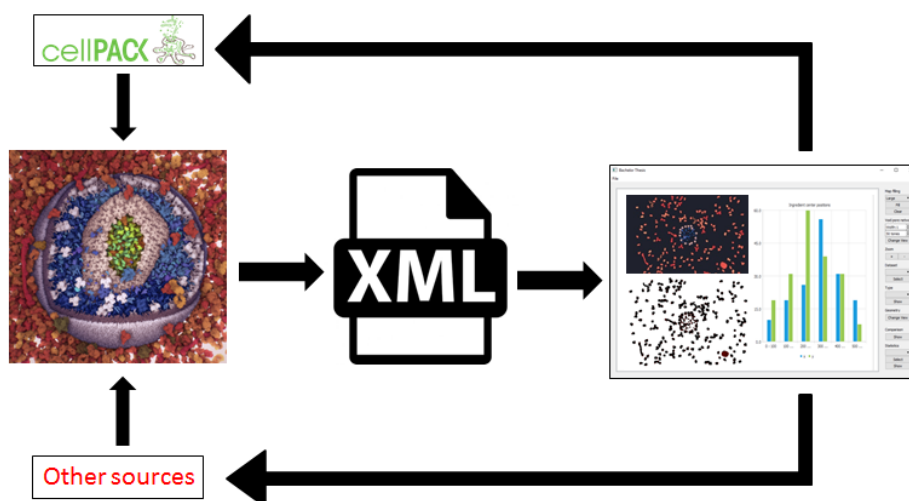


Figure 3.1: Flowchart representing the iterative process for the improvement of biological mesoscale models. These models can come from cellPACK or any other source that uses the same data (size and position of molecules and atoms). These data are converted into a single XML file that is later used by our evaluation tool, which enables the user to reach conclusions regarding the quality of the model, in order for them to change the model and restart the analysis process.

The main goal for this thesis is to create a software tool that can enable users to analyse models of the biological mesoscale. Those models can come from cellPACK or any other

source with the same data format. In order to be able to work with them, first the models need to be translated into written code files like XML or JSON. The reason for that is that we need a standardised data format that allows us to extract all the information about the model accurately and unambiguously. This is an efficient way to load and treat the data, which in our case will always describe 2D projections coming from 3D mesoscale models. Once it is accessed, the program will reproduce the visualization and offer the user a set of tools to evaluate visually the quality of the original model. The purpose behind this is to let users judge the compositions that the program receives as an input so they can modify and update the recipes to eventually create the best possible model for a given mesoscale biological system. This process is displayed in Figure 3.1.

3.2 Analysis tools

In order to offer the broadest range of possibilities for the evaluation, the techniques described in the related work section are taken into account and so the program includes the following functionalities:

3.2.1 Distance mapping and void space visualization



Figure 3.2: The distance of a point in the screen to the nearest object is showcased by colour gradient. In this figure there is an object placed right in the middle (fully white area) and every other pixel is coloured depending on the distance to the object.

One of the possibly desirable properties of an illustration is to have its elements homogeneously scattered around the screen. Like this, we get a better insight into the outer appearance and position of each ingredient, since elements packed too close together could potentially block the view to others and make details get lost in the crowd. Moreover, the natural randomness of biology makes it much more likely to find elements homogeneously placed. A map indicating a scattered distribution could point towards a better chance of dealing with a true-to-life model of the biological mesoscale. For this reason, our visual evaluation tool features a function that cellPACK authors[8] designated as 'void pore network'. It showcases the void areas of our 2D illustration by painting the screen

in different tones of red. Those areas that are occupied by an ingredient are shown white and those areas that are a given distance far from any ingredient of our dataset are painted in red. This distance can interactively be chosen by the user to suit the needs of their dataset. Any point that is not pure red or pure white will be shown in a combination of both colours depending on their distance to the nearest ingredient, with points becoming paler the closer they are to the nearest ingredient. An example for this can be found in Figure 3.2.

The input to create the void area map consists of the number of red tones and the width of each of the areas painted in a given tone. Both can be interactively chosen by the user. A small width and a big number of tones will result in a smooth transition between colours. The procedure is explained in Algorithm 3.1.

Algorithm 3.1: redDistanceMap

Input: A list *pixelred* with one array of points per red tone, two integers *numberredtones* and *redtonewidth* and an array of *object* elements *objects*

```

1 for  $x \leftarrow 0$  to screenwidth do
2   for  $y \leftarrow 0$  to screenheight do
3     mindistance  $\leftarrow$  screenwidth;
4     for  $z \leftarrow 0$  to objects.size() do
5       distance =
           $\sqrt{(\text{objects}[z].x() - x)^2 + (\text{objects}[z].y() - y)^2} - \text{objects}[z].radius();$ 
6       if distance < mindistance then
7         | mindistance  $\leftarrow$  distance;
8       end
9     end
10    for  $i \leftarrow 0$  to numberredtones do
11      if  $i * \text{redtonewidth} \leq \text{mindistance} < (i + 1) * \text{redtonewidth}$  then
12        | pixelred[i] << point(x, y);
13      end
14    end
15    if  $\text{numberredtones} * \text{redtonewidth} \leq \text{mindistance}$  then
16      | pixelred[numberredtones - 1] << point(x, y);
17    end
18  end
19 end

```

3.2.2 Two-by-two model contrast

If a packing problem has scarce solutions for a given container because of its small volume or the lack of different ingredients, to name a few reasons, the stochastic difference

between models coming from the same recipe could be unnoticeable at first sight. The same could happen when modifying a parameter with little influence on the packing. When this is the case, it is helpful to have a tool at our disposal that can highlight the differences between models. In our program we let the user have this possibility by displaying with bright colours those areas where there is a mismatch between two pictures while colouring in grey those where there is no change.

3.2.3 Interactive model view selection

When studying biological systems, it is essential to know how each of its components behaves. A living organism is the complex combination of diverse macromolecules that interact with each other and so the position and properties of said molecules is vital for the proper functioning of the system. Because of that, our visual evaluation tool features a function that allows the user to display only a certain kind of ingredient. The user can then analyse the whole model by studying first how its components have been packed and utilise any other functionality of the program on the new model that includes only the selected ingredient type.

3.2.4 Visualization of the spatial distribution of ingredients

Statistics is usually a source of quantitative information but in our case we will extract statistical data from our mesoscale models to produce a visual qualitative display of the distribution of the ingredients across the screen. Our program offers when required both a bar chart and a box plot that provides the information about the position of the ingredients. For the bar chart, the screen is virtually divided into sections for both the x- and the y-axis and the number of ingredient centre positions found in each section counted. Like this, if we are working with a perfectly homogenous composition then each bar of the plot will have the same length. If on the contrary the ingredients are densely packed in the middle, then we can expect the chart to look like a Gaussian function. The box plot does not work with sections but instead it gives information about the ingredient distribution by displaying where the median and the quartiles of the dataset are located.

Bar chart

The axes are divided into six equally sized sections. The number of sections was chosen to be suitable for the purpose of the chart, which we explained in the methodology chapter of this thesis. Then the program sweeps through the list of objects counting how many of their centre positions are in each section.

Box plot

For the box plot we need one list with the centre positions in the x-axis and one for the y-axis. These lists are then sorted by size to place the lower values first. Then we calculate the extremes, medians and quartiles of those lists. The extremes are simply the

first and last element of each sorted list. For the rest of values we will use the function shown in Algorithm 3.2.

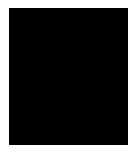
Algorithm 3.2: findMedian

Input: Two nonnegative integers *end* and *begin* and a sorted array of numerical values *list*

Output: The median value of the array

```
1 count  $\leftarrow$  end - begin;  
2 if count%2 = 0 then  
3   |   return list[count/2 + begin];  
4 else  
5   |   left  $\leftarrow$  list[count/2 - 1 + begin];  
6   |   right  $\leftarrow$  list[count/2 + begin];  
7   |   return (left + right)/2;  
8 end
```

The variables *begin* and *end* stand for a position in the sorted list. To obtain the median we will simply input the first and last positions, for the lower quartile we need the first and central positions and for the upper quartile the central and last positions.



Implementation

4.1 Pilot program

The implementation began with the creation of a basic program that could be later developed into the actual visual evaluation tool. For this matter, we worked with simple objects placed by the user and limited functionalities. At this stage, our objects have only two properties: their position on the screen and their size.

4.1.1 Selective placement of objects and automatic packing

When the mouse button is pressed while the cursor is inside the main window of the program, an object is created and automatically displayed in the position of the cursor. The size of the object is defined by a combo box where the user can choose from a list of values that offers a broad enough range for the purpose of this pilot program. In order to run tests on this program with numerous objects and a packed screen area, we added a push button that will fill the screen with objects when used. The program does so by sweeping the screen first from left to right and then from top to bottom looking for a spot where the biggest object possible can be placed, and repeating with smaller objects until no void spaces can be further used to pack objects (see algorithm 4.1).

4.2 Evaluation of cellPACK models

The program is designed to open XML files. The files that we work with contain the information needed to render a plane of a real 3D cellPACK model in a 2D illustration. The model consists of a number of macromolecules. In the XML file we find the following data about each macromolecule: the position and radius circle of its bounding circle, its type as a numeric value and the position of its atoms. Our software tool allows opening

Algorithm 4.1: autopacking

Input: An array of *object* elements *objects*, the positions *x* and *y* of the mouse cursor for both axes and an object size *z*

- 1 Repeat the following for every object;;
- 2 $dx = |x - objects[i].x|$;
- 3 $dy = |y - objects[i].y|$;
- 4 **if** $\sqrt{dx^2 + dy^2} < z + objects[i].radius()$ **then**
- 5 create new *object*;
- 6 add *object* to *objects*;
- 7 **end**

several xml files at once in the same window and selecting which one of them is displayed and analysed.

4.2.1 Two-by-two model contrast

When two different windows are open, we can use this tool to compare what we see on the screen. The function creates a new window by contrasting both images pixel by pixel and assigning a different colour to the new image's pixels depending of the result of that analysis. If a pixel is white in both pictures, the new one will be white. Otherwise it will be painted light grey for pixels that share the same colour and bright orange for those who do not.

4.2.2 Visualization of the spatial distribution of ingredients

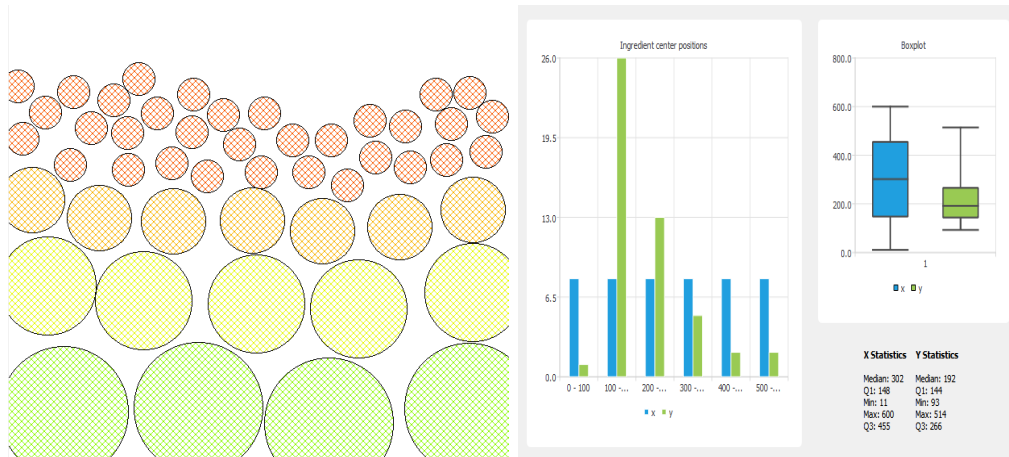


Figure 4.1: The chart on the right shows a homogeneous distribution along the x-axis of the portrayed model. The heterogeneity along the y-axis also becomes obvious.

The graphs needed for the visual evaluation have been implemented with the help of the

Qt Charts module. In this section we will demonstrate the effectiveness of the statistical tool of the program. For this matter, we manually placed a set of objects in order to achieve a homogeneous distribution along the x-axis and a strong gradient along the y-axis. As we observe in Figure 4.1, the bar chart for x values (blue) is even while the y values (green) show great disparity. The box plot reveals similar information: the blue box is longer, showing equal quartiles, the green one informs us of the fact that most objects are packed around the value $y = 200$ of the screen, which is the area on the top half where we find the small red objects. Finally, although the main focus of the program is on visual evaluation, the user also has access to the statistical data.

4.2.3 Void pore network

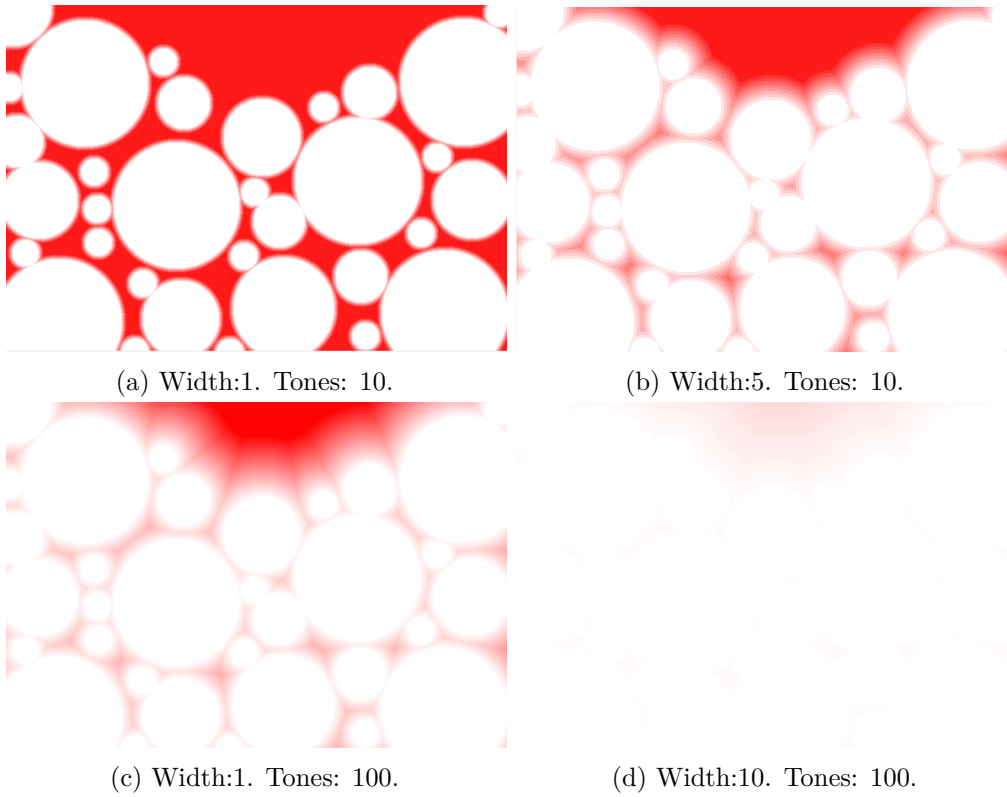


Figure 4.2: The program showcases the void areas of a model by calculating the distance of a pixel to the nearest object and painting darker tones of red for areas that are further away. The subfigures show different settings and combinations for the visualization of this property.

One of the first tools of the program tested was the creation of void pore maps. With the possibility of modifying the value of the number of red tones and the width of each tone, we got several results. As we can see in Figure 4.2, some combinations of both parameters can be useless as is the case of subfigures (a) and (d), where the map is either

just a color inversion of the original or an almost completely white picture. On the other hand, subfigure (b) has a rough transition between white and red but it would be the better choice if the screen was tightlier packed with no void areas like we see on top, because the small pores between objects would be much better visible. For the given example, subfigure (c) gives allows for the best visual evaluation. We can easily see the top void area while also realising that there is no collision between objects.

4.2.4 Interactive model view selection

By default, the program will show the models using the bounding circles of the macromolecules. However, the user can choose to display an atomic-detailed view. The global position of the atoms is not explicitly given in the xml file. Instead we are given the position \vec{p} of the bounding circle, the unit quaternion $\mathbf{q} = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ and the local position of the atom \vec{r} . To obtain the global position of each atom we need to solve first the following equation using the rotation matrix of the quaternion[14]:

$$\begin{bmatrix} xpos \\ ypos \\ zpos \end{bmatrix} = \begin{bmatrix} p1 \\ p2 \\ p3 \end{bmatrix} + \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2zw & 2xz + 2yw \\ 2xy + 2zw & 1 - 2x^2 - 2z^2 & 2yz - 2xw \\ 2xz - 2yw & 2yz + 2xw & 1 - 2x^2 - 2y^2 \end{bmatrix} \begin{bmatrix} r1 \\ r2 \\ r3 \end{bmatrix} \quad (4.1)$$

Moreover, the user has the possibility of viewing only one type of ingredient at a time, which is one of the goals explained in the methodology section. To achieve this, a new list of objects that includes only those of the selected type is created and displayed. Both selections do not only affect the illustration shown in the main window, it is also intended that the new lists of objects created by them work with every other functionality of the program like the charts or the void pore network. Like this we could for instance use the charts to analyse if there is a homogeneous distribution of the macromolecules but also change the view and analyse the distribution of the single atoms.

Results

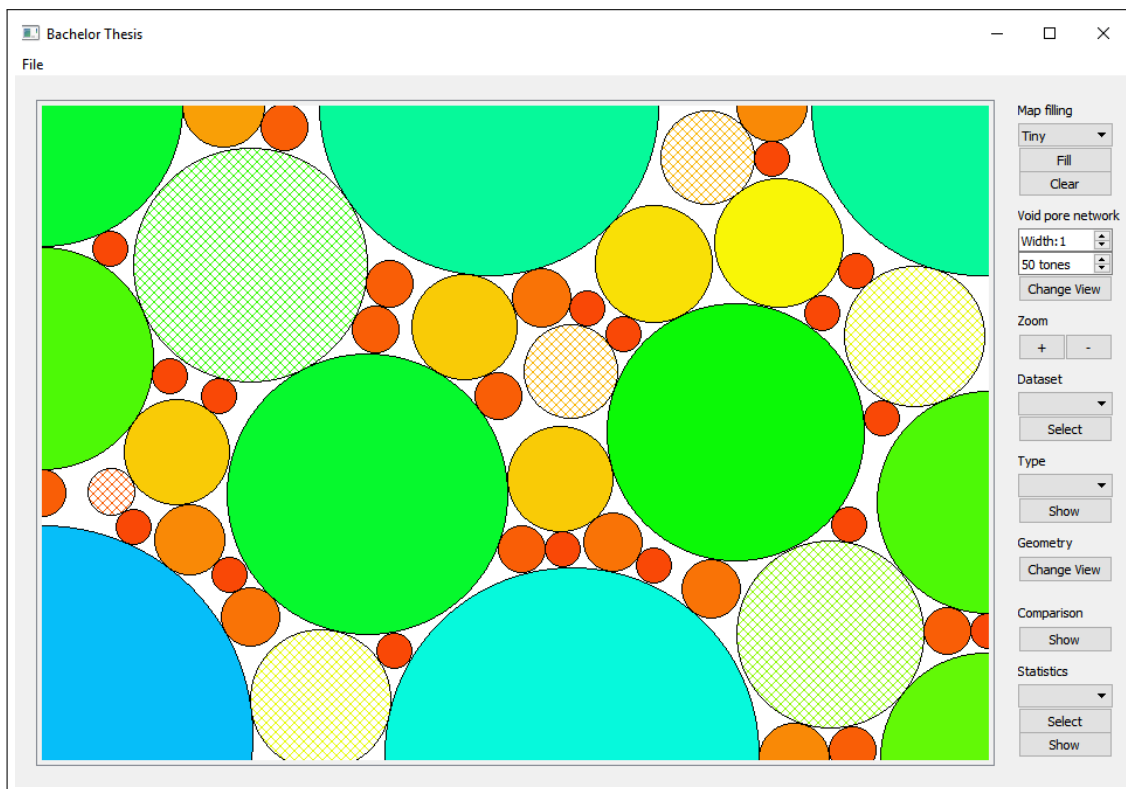


Figure 5.1: The graphical user interface of the program includes a main screen where different models can be displayed. This figure features a combination of manually and automatically placed objects.

The visual evaluation program in its latest version is displayed in Figure 5.1. Its simple design is composed by the main screen area where models are drawn and a sidebar that features all analysis tools as described in this thesis. In the figure, the main screen area is filled by objects, some of them placed manually and others created by the automatic packing function. To make the visualization more understandable, automatically placed objects are fully coloured while those manually placed have a diagonal cross pattern. Furthermore, the colour of the object is dependent of its size, with bigger objects showing colours with a smaller wavelength.

In order to test the main analysis tools of the program, we need to use real models. As previously mentioned, these models are converted into XML files and the program can load several of them at once. The models that we work with were created with cellPACK but our aim is to analyse compositions coming from other sources too. In Figure 5.2 we see on the left such models displayed with Unity3D. On the right there are two different views created by our program: the HIV virion is shown by painting the bounding circles surrounding its macromolecules, while the blood plasma is displayed in atomic level detail.

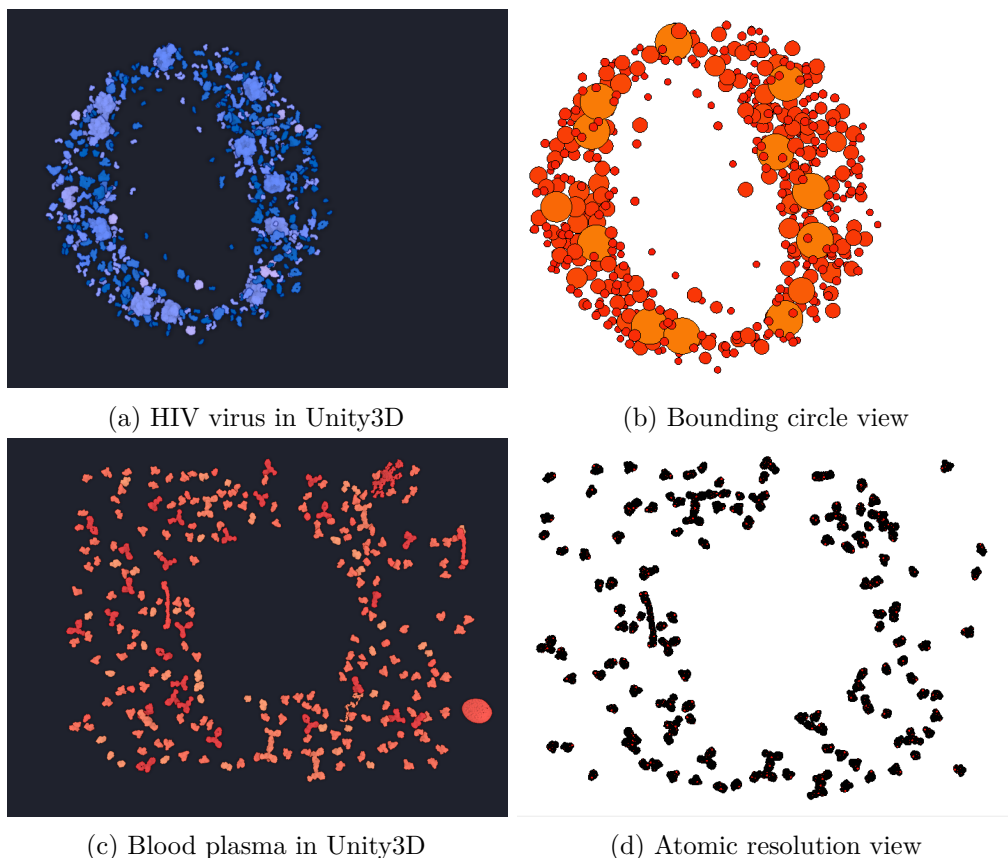


Figure 5.2: Comparison between cellPACK (left) and tool-generated models (right).

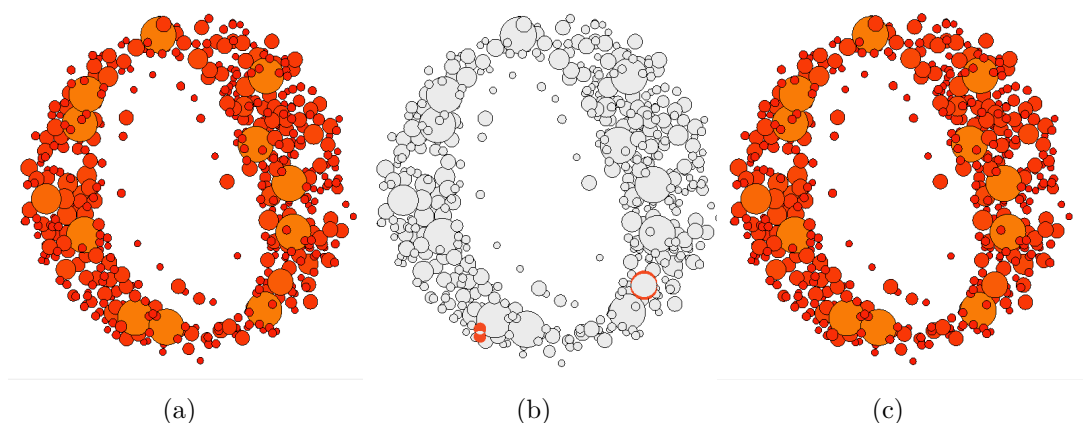


Figure 5.3: View (b) highlights the differences between models (a) and (c)

5.1 Two-by-two model contrast

Figure 5.3 shows one more possibility that the program has to offer: the two-by-two comparison. The difference between both models is two bounding circles that have been displaced, almost imperceptible to the human eye at first sight, but our software highlights it by painting the unequal areas in bright orange. This feature also works with atomic detailed resolutions.

5.2 Potential and limitations

In this section we will put the implemented analysis tools to the test. Figure 5.4 contains a set of pictures concerning an example dataset of a biological mesoscale structure. In this case we are dealing with blood plasma and the HIV virion put together in the same model. In the first row we see the original view and the recreation done by our program. The bounding circles view gives a better first impression of where the bigger and more complicated macromolecules are located, but both the original and the atomic resolution view prove that using bounding circles for long fibrous molecules distorts the illustration. In case the user wants a realistic recreation of the model, the atomic detailed view is recommended. The views with the bounding circles were useful at the beginning of this thesis because it enabled us to run the first tests on them without difficulties. In contrast, the atomic view meant working with thousands of objects. This lead for example to problems where the atomic view would take several minutes to load, which we later found out was caused by the code copying the full list with thousands of atoms instead of using pointers.

For subfigures (e) and (f) in Figure 5.4, a partial view of the model was used where only a certain type of molecule is represented. This is especially useful because most of the time we will want to know how a single kind of molecule is packed and how it behaves with the rest of the system. Studying the system as a whole is usually not the

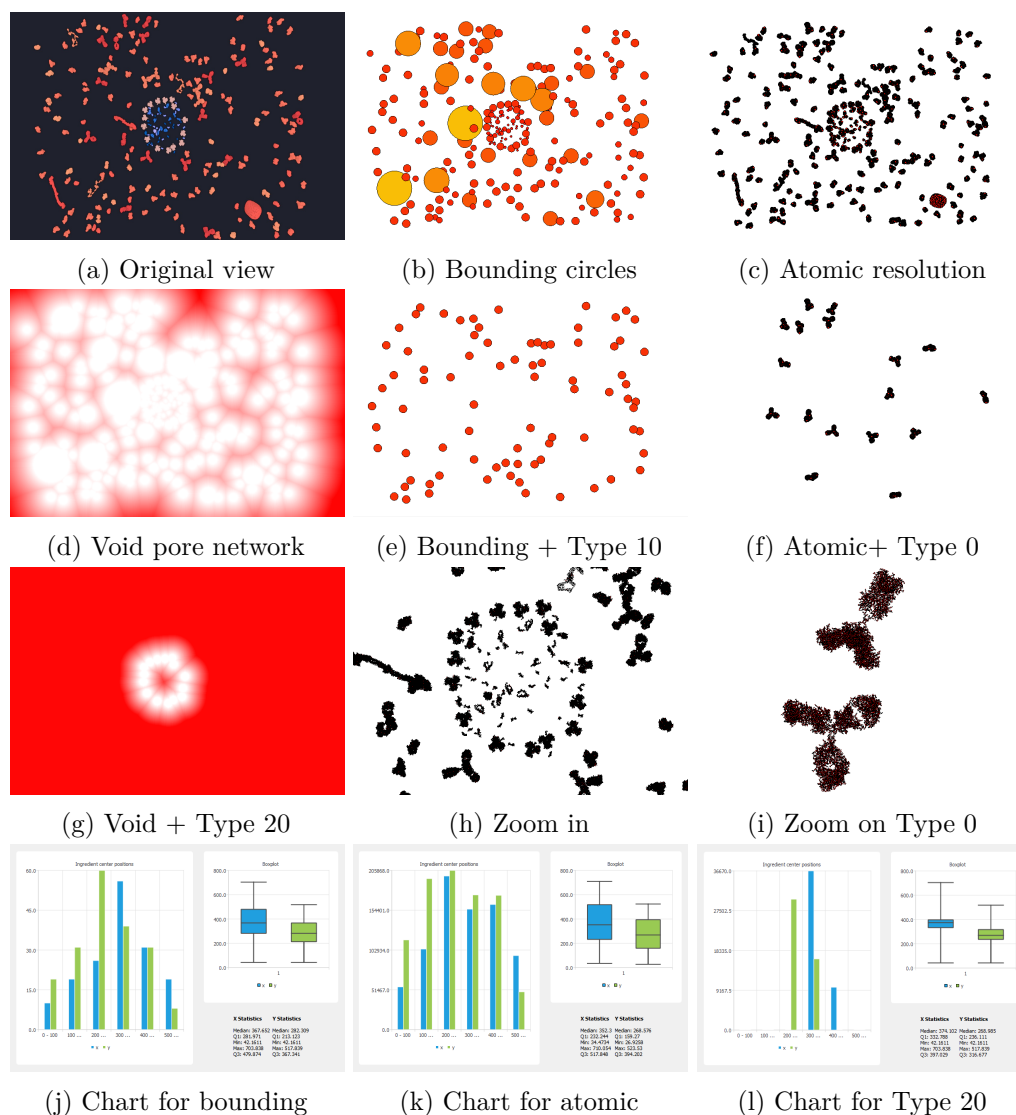


Figure 5.4: A proper evaluation of a given model must be carried out by combining the different views and tools offered by the program. The subfigures include each a stage of the analysis.

best approach due to its complexity. For instance, the chart found in subfigure (j) leads us to the conclusion that most objects are located in the centre of the screen, but this information may not be relevant for the user who can see that at first glance in (b). However, the chart for the atomic level detailed model in (k) shows a more homogeneous distribution of the composition and combined with the other chart we can conclude that the bigger and more complex molecules are found outside the centre area. One of the limitations of the chart function is that it cannot be modified interactively by the user,

and so as we see in subfigure (l) which contains the distribution chart for a certain kind of molecules which in this case were labeled as 'Type 20', if the objects are tightly packed in a certain area we need to use smaller chart sections or else we will lose information. In this case we learn that 'Type 20' molecules are located around the centre area, but we do not get to see that these macromolecules build a container. In (g) this is displayed clearly thanks to the void pore network. We can even see that the molecules do not form a perfect circle, but instead there are a couple of spots that we would expect to be white and are not. Void pore network is not the most powerful of the program but it might be the most versatile. It does not only work with isolated molecules like in (g) but also provides information of the whole composition as we see in subfigure (d). The main issue with the void pore network is the fact that currently it is not compatible with atomic level detailed views. This will be discussed in the next chapter. Subfigures (h) and (i) show the zooming functionality of the program, which enables the user to visualize the position of each atom in full detail. While (i) displays a perfect example of immunoglobulin molecules, (h) reveals one of the aspects where the program needs to be improved. As we can see in this subfigure in the model that we are working with in this example there is a perfect distinction between the spherical structure located in the middle and every other molecule on the outside. However, it is currently not possible to eliminate those molecules that we do not wish to visualize and instead the user is limited to either show all molecules or only a certain type.

Conclusion and outlook

The program designed and implemented in this thesis is a first step towards a thorough and multifunctional tool for the visual evaluation of illustrative models of the biological mesoscale. Currently it provides a limited number of functions with which users can do a basic analysis of assembled structures, although the intention is to further develop the program to offer more possibilities and improve those that we already have.

The previous chapter explaining the results of this thesis proves that we provide several combinations of tools and views that can be used for visual evaluation of models. It is up to the user to interact with the software and utilise those tools that are most suitable for their model. For instance, the tool that provides a two-by-two comparison of models may be suitable for exceptional cases where we are dealing with almost undistinguishable models but often such models are different enough for the tool to provide pointless information. In any case, in the future it would be desirable to further simplify the program so it is more intuitive and accessible to all audiences, and offer guidance and help boxes through the process of analysis.

As previously mentioned, there are some points where we suggest improving the program. An example for that is the lack of possibilities to work with several types of molecules at once. For this purpose, we propose not only changing the code but also adding options so the user can interact with the program more easily with the use of mouse dragging and clicking instead of relying so heavily on push buttons and combo boxes. Also, it would be desirable to be able to study the interaction and relationship between ingredients, which could possibly be a whole new functionality in the program.

One further limitation of the program is displaying the void pore network of the atomic resolution view. As it is now, the algorithm that creates it is too computationally expensive when combined with a large amount of objects and eventually fails when used. A suggested solution is to change the method of the void pore network calculation to avoid using the objects list. Instead we could calculate it by using the colour information

of each pixel, with white meaning void and any other colour meaning filled. Then we could start an iterative process to produce a soft transition between red and white areas, with every pixel checking the colour property of the surrounding ones.

Finally, we should not forget that this tool is thought to help researchers evaluate the quality of mesoscale models. Such models could come from the most diverse sources and so the program needs to be able to work with more than just a certain XML file format. For the sake of completeness, we propose to add a database in the future where information about mesoscale structures and previous evaluations is compiled in order for the program to automatically recognize molecules and patterns, and give guidelines on what the user is seeing on the screen. Last but not least, one of the main developments in this area would be to provide analysis tools for 3D models as a whole and not limit ourselves to the evaluation of 2D slices which is obviously not the best approach. In any case this exceeds the expectations of this thesis and we conclude by stating that the goal of providing users with a basic tool for visual evaluation of computational models of the biological mesoscale has been successfully achieved.

Bibliography

- [1] Bo Huang, Hazen Babcock and Xiaowei Zhuang, *Breaking the Diffraction Barrier: Super-Resolution Imaging of Cells*, Cell. 2010; 143(7): 1047–1058.
- [2] Gregory A Petsko and Dagmar Ringe, *Protein Structure and Function*, New Science Press 2004, Chapter 5.
- [3] Modelling the biological mesoscale: an interview with Professor Art Olson, <https://goo.gl/hwm5Jd> News Medical, 2015.
- [4] University of Utah, Genetic Science Learning Center, *Cell Size and Scale*, <http://learn.genetics.utah.edu/content/cells/scale>
- [5] Wikipedia. Computational model — wikipedia, the free encyclopedia., https://en.wikipedia.org/wiki/Computational_model Accessed: 2017-06-15.
- [6] Graham T Johnson, David S Goodsell, Ludovic Autin, Stefano Forli, Michel F Sanner and Arthur J Olson, *3D molecular models of whole HIV-1 virions generated with cellPACK*, Faraday Discuss., 2014, 169, 23-44.
- [7] Wikipedia. Packing problems — wikipedia, the free encyclopedia., https://en.wikipedia.org/wiki/Packing_problems Accessed: 2017-06-15.
- [8] Graham T Johnson, Ludovic Autin, Mostafa Al-Alusi, David S Goodsell, Michel F Sanner and Arthur J Olson, *cellPACK: a virtual mesoscope to model and visualize structural systems biology*, Nature Methods, 2014.
- [9] cellPack, <http://www.cellpack.org> Accessed: 2017-06-15.
- [10] Tobias Klein, Ludovic Autin, Barbora Kozlíková, David S. Goodsell, Arthur Olson, M. Eduard Gröller and Ivan Viola, *Instant Construction and Visualization of Crowded Biological Environments*, IEEE Transactions on Visualization and Computer Graphics. (To appear in 2017).
- [11] Madhura N. Phadke, Lifford Pinto, Oluwafemi Alabi, Jonathan Harter, Russell M. Taylor, II, Xunlei Wu, Hannah Petersen, Steffen A. Bass and Christopher G. Healey, *Exploring Ensemble Visualization*, SPIE-The International Society for Optical Engineering 8294(82940B), 2012.

- [12] Muhammad Muddassir Malik, Christoph Heinzl and M. Eduard Gröller, *Comparative Visualization for Parameter Studies of Dataset Series*, IEEE Transactions on Visualization and Computer Graphics, Volume 16, Issue 5, 2010.
- [13] Ismail Demir, Christian Dick and Rüdiger Westermann, *Multi-Charts for Comparative 3D Ensemble Visualization*, IEEE Transactions on Visualization and Computer Graphics, Volume 20, Issue 12, 2014 .
- [14] Wikipedia. Quaternions and spatial rotation — wikipedia, the free encyclopedia., https://en.wikipedia.org/wiki/Quaternions_and_spatial_rotation#Quaternion-derived_rotation_matrix Accessed: 2017-06-15.