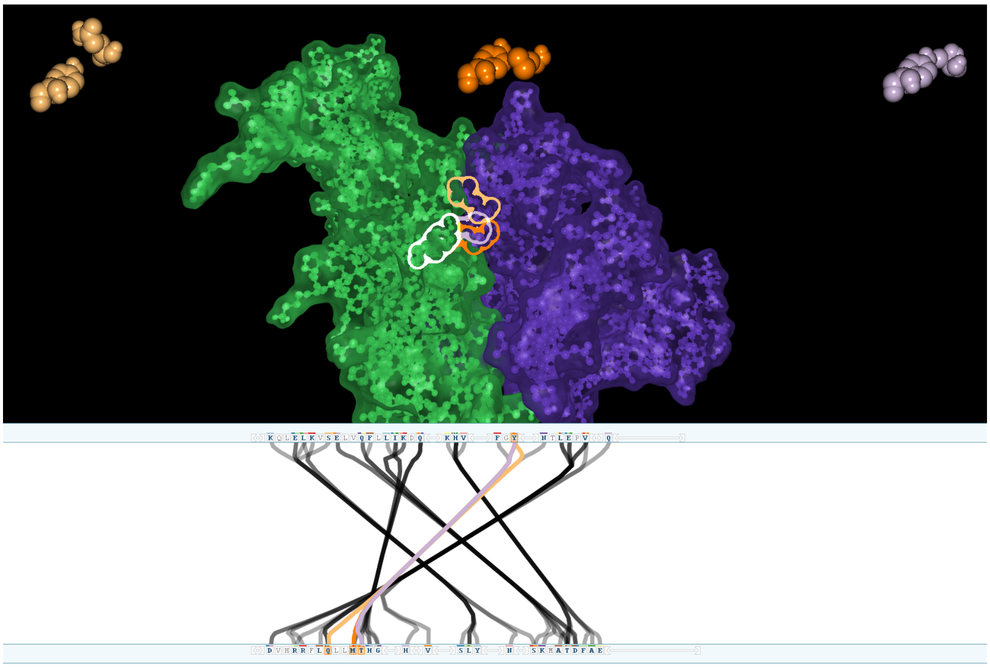
Visual Analysis in Protein Protein Interactions

Marius Tendeland Horne, author

Stefan Bruckner, supervisor

Master Thesis

University of Bergen
Department of Informatics


K Q L E L K V S E L V Q F L L I K D Q
K H V
F G Y
N T L E P V
Q
D V H R R F L Q L L M T H G
H
V
S L Y
N
S K M A T D F A E

Abstract

Over the last decade there has been a steady increase in the focus of research into Protein-Protein docking. The Docking software provides a plausible configuration to a Protein-Protein Interaction. The docking will also provide analysis and ranking of said plausible configuration of Protein-Protein Interaction. The Docking softwares are getting more reliable, but there are still parameters that the software can't handle, and domain experts have to manually explore the configurations to find and select the relevant ones, which is a time consuming process. With our software, the time used by domain experts to explore the configurations, will be reduced. This software provides a nice overview of the connections between the two proteins in a Protein-Protein Interaction, and provides 3D visual aid to locate the spatial orientation of the contact zone in the proteins, and the Amino Acid pairs in the contact zones spatial orientation to each other.

Acknowledgements

Thanks to Stefan Bruckner for supervising and helping with this project and coming up with good solutions to work with. Thanks to Jan Byška for many helpful discussions and pointers on how to use Caver and information on what Protein Protein Interactions are and why they are useful. I also want to thank Adam Jurčík for helping with how to use the render functionality in Caver.

Contents

Chapter 1

Introduction

Proteins are in every living organism and the proteins are the cause of most major biological processes, such as muscle contraction, cell signaling, cellular transport, biochemical pathways, immune system and cell division. In short, proteins are directly involved in the chemical processes essential for life. Therefor, understanding the functionality of proteins are crucial. The functionality of proteins are tightly connected to how proteins are bound to other proteins in a Protein-Protein Interaction. When two or more proteins interact, there are several measurable effect which can occur :

- Alter the kinetic properties of enzymes.
- Creating a new binding site.
- Inactivate or destroy a protein.
- Provide a functionality, which neither protein can exhibit alone.

Knowledge of how proteins work are of great interest to research disciplines such as medicine and pharmaceutics. This is because the knowledge of this can be put into drug-design systems to develop better drugs and also for therapeutically use, such as help with cancer, and better understanding of human diseases. But the understanding of Protein-Protein Interactions are lacking. One of the reasons for this is the nature of some of the Protein-Protein Interactions. The interactions can be divided up into stable or transient interactions. The stable interactions are easier to purify and identify with wet science methods, such as screening, x-ray crystallography or mass spectrometry. The problem is that most cellular processes are controlled by the transient interaction and the chaotic nature of these transient interactions makes it harder to prepare samples and purify, so that the interactions can be identified. Aside from the problems wet science is having with transient interactions, it is also expensive to do this.

With the problems regarding wet science, there is a need in the community that study these biological structures to still achieve feasible proposed configuration of a Protein-Protein Interaction computationally. A configuration is a set of proposed Amino Acid pairs that connect the proteins together. The process of producing said feasible configurations is Protein-Protein Docking. The goal of Protein-Protein docking is to create a prediction of a protein-protein interaction configuration as it would occur in a living organism. New tools and algorithms to create these predictions has appeared in later years. There still remains a problem, as the Protein-Protein Docking only provides feasible predictions of configurations and they provide a large number of possibilities.

Domain experts will have to explore these configurations manually, to look for and select the configurations which are biochemically relevant to their work. To reduce the number of configurations a domain expert has to explore, most of the docking softwares provides a ranking of the configurations. A ranking is provided with utilizing methods such as scoring functions to identify structures that would most likely occur in nature. The scoring functions consider several parameters such as hydrophobicity, surface area, spatial cluster, geometry of the surface, ect. The Critical Assessment of PRedicted Interactions (CAPRI) are running rounds for docking softwares to test how accurate their predictions are.

With the smaller set of configuration left after the ranking, the domain experts still need to explore the reduced set of configurations manually. Here visual support is essential, as this will let them compare the differences in configurations and look at the spatial orientation of contact zones. A contact zone is a set of amino acids from both interacting proteins that are in a given configuration. Visualizing these contact zones in 3D proves to be a problem, as the contact zones are often between the two proteins, and thus there are multiple Amino Acids surrounding the contact zone and obscuring said zone.

It is with this last step, to visually help domain experts explore and select configurations, our software is meant to help. We provide a tool to achieve a overview of the Contact Zone with a 2D graph based visualization. By selecting Amino Acid pairs from the contact zone with the help of the 2D visualization, we provide a way to look at the connection in 3D. The 3D view will provide a visual aid to see the spatial orientation of the contact zones, and also the spatial orientation of the Amino Acids in a pair to each other.

1.1 Thesis Structure

- Chapter 1 gave an introduction to what Protein-Protein Interaction is, where we get our configurations from and why we need our software.
- Chapter 2 will present what others have done in the same research area as this thesis belongs. We will start by explaining Molecule Visualization, as much of this Thesis uses the molecular visualization as a ground base(our 3D views). Furthermore we will look at what the focus are in Protein Protein Interaction networks and also look at what others have done to visualize the interaction between two proteins.
- Chapter 3 is were we present out methodology and some questions we want a user to be able to answer after using our software. We will explain how we created our Selection Tool and our two 3D views.
- Chapter 4 will go through the techniques that are not straightforward to implement, which are needed to construct the concepts from Chapter 3.
- Chapter 5 will contain results, to show that a user can answer the questions proposed in Chapter 3.
- In Chapter 6 we will look at the limitations with our software and discuss changes.

- Chapter 7 will contain a conclusion and what we can do to improve this software.

Chapter 2

Related Works

In this chapter we are going to look at existing work in molecular visualization and take a closer look at what is being done in visualization of Protein-Protein interaction. We will look at both visualization of multiple Protein-Protein Interaction or what is called a Protein-Protein Interaction Network, and methods to visualize two proteins in a Protein-Protein Interaction.

2.1 Molecular Visualization

This branch of visualization has been around for quite some time. It has deep roots in hand drawn visualization, but due to larger and larger datasets, the need to be able to produce more accurate images in a less time consuming manner became more and more crucial to get an understanding of all the new data. Because the molecular data is varied, it arose the need for different types of visualizations and the two types can be categorized as Atomistic Model rendering and Illustrative / Abstract rendering. Kozlíkova et al[KKF^{+}16] has a thorough State of the Art on Molecular Visualization, a great deal of what we have here is from their state of the art. With regards to some visualizations such as Surface rendering with both abstract and atomistic, we haven't gone as deep as they have. What we have done instead in this thesis, is go through the general differences in rendering of molecular visualization as available in the 3D section in our software.

Atomistic Model

Atomistic Models is the rendering where you visualize the data of each individual atom in a molecular complex. Atomistic models can be divided into two main groups, the "Bond-Centric" models and "Surface" models

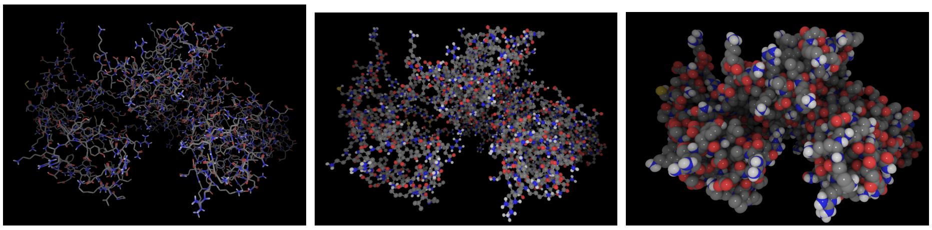


Figure 2.1: First figure is Sticks, then ball and stick, last is Van der Waal spheres

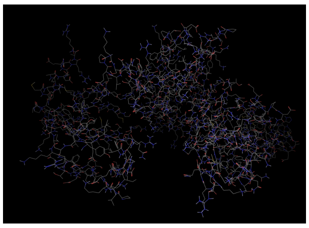


Figure 2.2: Image of the wireframe technique (lines)

Bond-Centric models are rendered with focus on showing the bonds between each atom. This can help with understanding the chemical properties of the molecule by being able to see the placement of atoms and / or the connections between the atoms. One of the most frequently used method is the Stick figure. The Stick figure is where you render the bonds between each of the atoms, this can again be augmented with showing the atoms on top of it. This is called Ball and Stick, and it resembles the scientific toy, students use to build together molecules. One of the simpler way to render it is with wireframe (lines), as shown in figure 2.2, of the complex, where you draw a line for the bonds between each atom. When it comes to rendering the atoms, a sphere is the most popular representation of the atom. The radius of the sphere is given with the Van der Waals Radii, which is a radius representing the distance of the closest approach to another atom. This kind of spherical representation with the Van der Waals radii is called a Van der Waals sphere / balls. Van der Waals balls are quite often used, beside used as a representation they are also used as a basis for some of the surface rendering methods, Figure 2.1 shows the stick, ball and stick and the Van der Waals representations.

Surface rendering is used to render the surface of a molecule. The most straightforward way to render a surface is with the Space-Filling method, which is where the surface is the union of all the surfaces of the atoms in the molecule. If you use the Van der waal spheres instead, and take the union of the surfaces of the atoms using the Van der Waals radii, you get the Van der Waals surface. The Van der Waals surface is the surface which has been most expanded upon, the first one was the Solvent Accesible Surface(SAS) which were proposed by Lee and Richards[LR71]. The idea here, was to show a surface that would visibly display all the regions, where solvent molecules has access. To achieve this they would start with a Van der Waal surface, and they would have another sphere, the probe. The probe would then roll over the Van der Wall surface, and you would create a new surface of where the center of this probe were at all times. This form of visualization has the benefit of showing all areas where a solvent molecule of the same radius or less than the probe sphere has access. With this it became feasible to visualize and analyze possible binding partners. The problem with this visualization, is the surface itself. The surface you would have visualized would not be accurate. As another possible surface to SAS the Solvent Exuded Surface(SES) was made. The SES compared to SAS uses the surface of the interaction point with the probe and Van der Waal surface instead of the probes center. SES retains the properties of SAS to visualize the possible binding partners and transport channels, and it retains a more accurate surface model of the molecule complex. After these two surfaces there has been done heavy research into making more accurate surface models and also have them be real time interactive. Krone et al [KBE09] proposed a dynamic and interactive representation of the SES surface with the use of GPU ray casting techniques. Surface rendering is likely to be one of the most popular ways to render Protein-Protein Interaction between two proteins, as they get the details from SES and can later use enhancement techniques to visualize the contact zones.

Abstract Model

Larger molecular complexes rendered with atomistic representations will have a certain degree of occlusion by the cluttering of large set of atoms and bond between the atom. This was one of the reasons abstract model representation of molecules came to be. The other reason was that one would not always need the information provided by the atomistic models, but would rather like to better see the sub-structures or specific features of the molecule complex. One of the most known abstract models for molecule rendering is the visualization of DNA and RNA as a step like ladder. Cartoon rendering is also an abstract model for rendering molecular complexes. With the cartoon visualization, the secondary structures are visualized as ribbons and arrows. This is often used to visualize Protein-Protein Interaction, where one of the proteins are rendered as a surface, and the other interacting protein is rendered as ribbons. This is done to visualize the hot spots in the interaction. Where the hot spots are the residues which can change the affinity of the interaction with being mutated.

2.2 Molecular Enhancement

While much research has been done to figure out possible ways to visualize the molecule with different models from atomistic to abstract, there is also a need to be able to enhance the molecular rendering with complementing info from the data of the molecule. Color is a way to enhance the molecule to show distinct data. One can either render each of the atoms by their type, color by chains, function units, bonds, hydophobicity and other criteria one would wish to show in their rendering. Ambient Occlusion is a technique that is made to mimic the diffuse light between objects. This gives the molecule localized shadows in creases, which helps with the depth perception. Another way to improve on the depth perception is Halo rendering, which is a technique where you rely on the human capability of noticing small changes in contrast. The Halo can be drawn around a chosen object as a dark outline for depth, or it can be used as a glow to help detach the chosen object from the rest.

2.3 Protein-Protein Interaction

In this section on Protein-Protein Interaction we will focus on two different styles of visualizing. The first one is graph based visualization of multiple proteins in a PPI

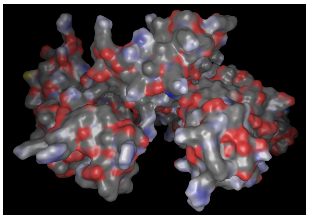


Figure 2.3: Surface representation of the structure.

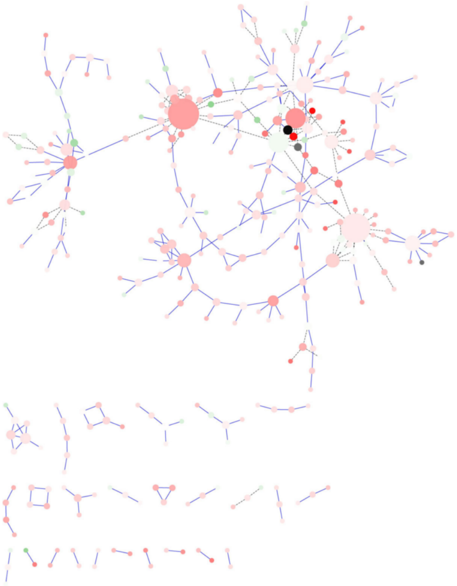


Figure 2.4: **Visualization of a network using Force Directed Algorithm in Cytoscape.** Visualization of pathways of galavtose (galfiltered) network usinf Force Directed Algorithm in Cytoscape. Figure is from [AGC13].

network and secondly we will focus on the visualization between two proteins in a PPI.

Protein Interaction Network

To understand biological processes such as immunity and metabolism, Protein Interaction Networks must be understood. Sevimoglu et al[SA14] has done a review on how work on Protein Interaction Networks is important for Biomedicine, with more understanding we would better know the human diseases and in the future would be able to provide more personalized strategies for controlling the disease pathways.

The Protein Interaction Networks consist of multiple proteins and a large number of connections, therefor work on visualizing this subject has been done as graph representation. In the graph representing the Protein Interaction Network one uses the proteins as nodes and the interactions as edges. Graph theory is a well developed field within the Visualization. Agapita et al[AGC13] has done a thorough report on how different 2D graph layouts can be used with regards Protein Interaction Networks.In this report they also go through softwares implemented to visualize these large Protein Interaction Networks, and they defined a set of main requirements as following : "

- Clear rendering of network structure and substructures, such as dense regions or linear chains;
- Fast rendering of huge networks;
- Easy network querying through focus and zoom; Compatibility with the heterogeneous data formats used for Protein Interaction Network representation;
- Interoperability with PPI databases, allowing the automatic querying of single or multiple databases using existing middlewares;

- Integration of heterogeneous data source;

" They found that there are multiple free, open-source tools such as Cytoscape[SOR+11], which is a popular bioinformatics package for biological network visualization and data integration. Cytoscape allows for both 3D and 2D visualization of protein interaction networks, making layout algorithms available. Some tools are integrated and let you chose a data base to receive data from, and some of the tools specialize in the specific use of analysis of pathways and cellular processes.
As shown with Agapita et al, there exist a large amount of research into layout regards to Protein-Protein Interaction networks. An example to layout algorithms are the Hive-Plot proposed by Krzywinski et al[KBJM12]. Their hive-plot used node coordinate system rather than force-directed layout as is quite common to use in PPI network layouts, which gives the networks a consistent layout and makes it possible to compare multiple networks. As Krzywinski et al proposed a layout which would let a user compare multiple networks, Singh et al[SXB07], developed a Parivise ranking with Iso matrices of two PPI networks. Here they match the networks by matching proteins together, only if their neighbors also match each other.

Jeanquartier et al[JJQH15], did a similar research as Agapita et al. While Agapita et al focused on softwares, Jeanquartier et al focused on web visualization which were integrated with databases. They found out there is still room for improvement with the interactivity in the tools as most tools lacked some of the exploration techniques they looked for. Further enhancement of visualization with regards to biochemical analysis tool.

Protein-Protein Interaction

When it comes to PPI networks, the work has been more centered around the layout of the graphs to be able to discern sub-units in the network. But with Protein-Protein Interaction between two proteins, the focus will go more towards techniques to display the connections in a configuration or to enhance visualization of a molecular rendering to provide visibility to the surface area and / or the contact zones of the configuration.

Laskowski et al[LS11] expanded on already existing technology in LigPlot, which generates schematic diagrams of protein-ligand interaction for a given PDB file. In this they have superimposed the 3D coordinates of the protein-ligand connections, into a 2D position in a diagram. The diagram shows the hydrogen-bond and hydrophobicity between the ligands and the main-chain in the protein. With this possibility and the ability to have multiple protein-ligand interactions visible at once, this lets people analyze a series of small molecules which binds to the protein.

Sansen et al[STDB16] proposed a software to visualize the sRNA-mRNA Interaction prediction. The need for this, is that in later years it has come into light that proteins are created by non-coding RNA (sRNA) coupled with the mRNA, so by exploring the connection between them they can better understand proteins. The way that this interaction pair is visualized is by first drawing the sRNA secondary structure and then the mRNA are placed around the secondary structure and is then coupled to the secondary structure.

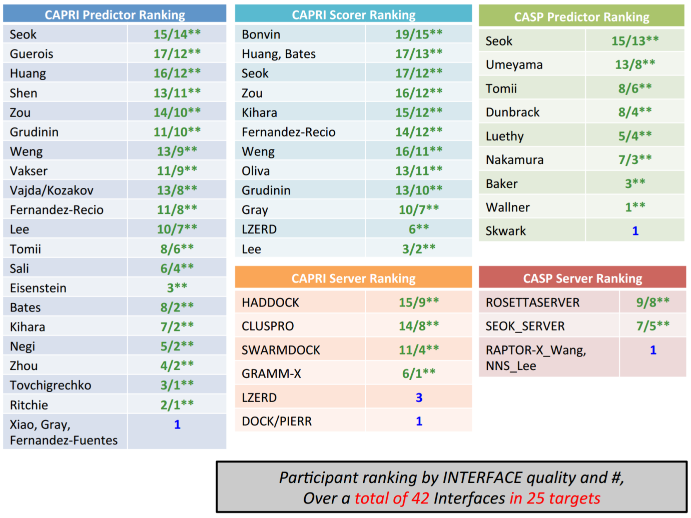
Ban et al[BER04], decided to take an geometric approach to visualize the interaction in a protein-Protein Interaction. They proposed to visualize the geometry of the interface surface formed by the two proteins. With the use of a space-filling diagram they developed, they visualize the interface areas on both proteins as a single entity as a geometric shape. With this as an entity by itself, they envision it will help with better understanding and exploration of a Protein-Protein Interface with regards to its physiochemical abilities and the interacting pairs to each other.

Chapter 3

Methodology

The Critical Assessment of PRedicted Interactions (CAPRI) is a community experiment where different docking softwares can attend to test the accuracy of their predictions generated by their software, by doing a blind test against unpublished experimental structural complexes. This is based on the willingness of structural biologist to provide unpublished experimental structures as targets. These targets comes from techniques such as crystallography and NMR spectroscopy.

Figure 3.1 is a scoring table taken from CAPRI round 30[LVKW14], where the participants who made the best predictions on the given experimental structures are on the list. The predicted configurations that were accepted where the ones who were above the grade of "acceptable", which means that at least more than 10% of the Amino Acid pairs in a configuration were correctly predicted. The ones who shows double stars ("**") next to the number i.e "19/15**" shows that 15 were of "medium" grade, which means that at least 30% of the Amino Acid pairs were predicted correctly. But there are no rankings who received three stars("***"), which would be 50% of the pairs correctly predicted. Which shows that even with the increase in Docking computation and predictions, there will still be configurations which will not have all pairs correctly predicted, and therefor we still need domain experts to manually explore the



CAPRI Predictor Ranking	
Seok	15/14**
Guerois	17/12**
Huang	16/12**
Shen	13/11**
Zou	14/10**
Grudinin	11/10**
Weng	13/9**
Vakser	11/9**
Vajda/Kozakov	13/8**
Fernandez-Recio	11/8**
Lee	10/7**
Tomii	8/6**
Sali	6/4**
Eisenstein	3**
Bates	8/2**
Kihara	7/2**
Negi	5/2**
Zhou	4/2**
Tovchigrechko	3/1**
Ritchie	2/1**
Xiao, Gray, Fernandez-Fuentes	1

CAPRI Scorer Ranking	
Bonvin	19/15**
Huang, Bates	17/13**
Seok	17/12**
Zou	16/12**
Kihara	15/12**
Fernandez-Recio	14/12**
Weng	16/11**
Oliva	13/11**
Grudinin	13/10**
Gray	10/7**
LZERD	6**
Lee	3/2**

CAPRI Server Ranking	
HADDOCK	15/9**
CLUSPRO	14/8**
SWARMDOCK	11/4**
GRAMM-X	6/1**
LZERD	3
DOCK/PIERR	1

CASP Predictor Ranking	
Seok	15/13**
Umeyama	13/8**
Tomii	8/6**
Dunbrack	8/4**
Luethy	5/4**
Nakamura	7/3**
Baker	3**
Wallner	1**
Skwark	1

CASP Server Ranking	
ROSETTASERVER	9/8**
SEOK_SERVER	7/5**
RAPTOR-X_Wang, NNS_Lee	1

Participant ranking by INTERFACE quality and #,
Over a total of 42 Interfaces in 25 targets

Figure 3.1: The scores from CAPRI round 30[LVKW14]

configurations.
To help with the manual validation of a configuration, we want our software to help give answers to these 5 questions[1] :

- Q1 : Which configurations contain a selected interacting pair of amino acids and what is the frequency of occurrence of this pair in all configurations?
- Q2 : Which pairs of amino acids are present in a given configuration?
- Q3 : How close are the amino acids in the contact zone and which are the closest ones?
- Q4 : How similar and different are the contact zones in the configurations?
- Q5 : What are the differences between the sets of amino acids in the contact zones of configurations?

Further on we will give a quick overview of the three parts of our software. These parts are the Selection tool, the Outline visualization and the Image in Image visualization. After the overview, we will look more closely on each of them and explain how they are made.

The contact information for which Amino Acid pairs which are the interacting pairs, are not generated in this software. The data is done with generating it by utilizing Protein-Protein Docking software, such as ZDOCK, HADDOCK or other Docking softwares.

3.1 Conceptual Use

The Selection consist of the symbol sequence of the two proteins, and a graph area with edges connecting to the symbols representing the Amino Acids in the connections given by the configuration. The reason we chose to represent an overview of the connections in this way, is the sequences will give a context to where an amino acid is in its protein, and the edges in the graph will show which other amino acids it is connected to. With this view it will help with finding out which interacting pairs of amino acids are present in a configuration, what the frequency of a selected pair is and also helps with comparing multiple configurations. Later, When we are taking a closer look at this tool, we are going to go through how one was made, but it is intended to make one of these for each configuration loaded. This tool also is mentioned first, since it is the way you use this which will affect the latter methods. With this, a user will be able to explore multiple or single connections by selecting an edge, which represents one pair, or a symbol (belonging to an amino acid) in the sequence, which selects all pairs which the Amino Acid belongs to. With multiple configurations, each of these selection tools will be aligned vertically to each other.

With the selections chosen in the Selection Tool, this visualization will let you either get a glow around the selected amino acid or a contour. This will help locate where the pair(s) are in the spatial location in the Protein Protein Interaction. This will help with giving an idea of how close the Contact Zones in the proteins are to

[1] These questions were, by permission, taken from a unpublished paper by Furmanová et al.[FBG+]

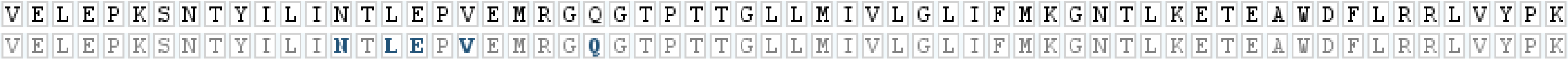

V E L E P K S N T Y I L I N T L E P V E M R G Q G T P T T G L L M I V L G L I F M K G N T L K E T E A W D F L R R L V Y P K
V E L E P K S N T Y I L I N T L E P V E M R G Q G T P T T G L L M I V L G L I F M K G N T L K E T E A W D F L R R L V Y P K

Figure 3.3: Shows sequence of a protein. The top shows sequences with no differentiation on amino acids which are in a connection in the Protein Protein Interaction, the bottom shows where we differentiate with those who are in a connection and those who are not.

each other and also how the Contact Zone is placed in the protein.

This last part is an extra option to add to the 3D visualization. This functions as an extra image on the rendering, showing a single Amino Acid Pair unobstructed. With this you can see how the pair is positioned and rotated to each other, as well as the distance between them. If multiple Amino Acid Pairs are chosen, the images will be spread around the border of the screen.

3.2 The Selection Tool

In this section we will describe how a Selection Tool for a configuration is created, if multiple configurations are loaded there will be one Selection Tool for each. In the case of multiple Selection Tools created, the Tools are aligned vertically to each other in a window. With this option of multiple Selection Tools beneath each other it will make it simpler to see the similarities of the configurations. We have divided this section in two parts, where the first one will focus on the sequences of the proteins, and the second part will focus on the connection graph which connects the sequences.

Sequence

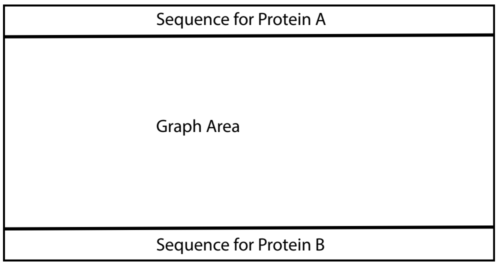


Figure 3.2: A illustration of the areas which the Selection Tool consist of.

When we refer to a sequence in this thesis, we think of a visual representation of the amino acids in the proteins chain. The sequence consists of the symbols of a corresponding amino acid, i.e Alanine has the symbol A, Glutamine has the symbol Q and so forth. In the configurations we work with, we handle two proteins at a time. We have a sequence of the amino acids in the protein for each of the proteins. We have separated the sequence from each other, and have placed them vertically to each other with space for the connections in between. Figure 3.2 illustrates how the area of the Selection Tool will look.

In our sequences, we would like to locate where in the sequence the Amino Acids which are a part of the contact zone are. When we are able to locate where the "important" Amino Acids are in the sequence, we can tell which Amino Acids are present in a given configuration, and also gain information on where in the chain

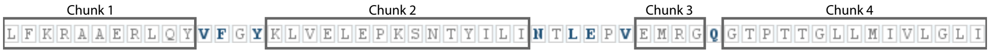

Chunk 1
Chunk 2
Chunk 3
Chunk 4
L F K R A A E R L Q Y V F G Y K L V E L E P K S N T Y I L I N T L E P V E M R G Q G T P T T G L L M I V L G L I

Figure 3.4: Shows the chunks which are to be abbreviated.

they are located, if they are close together or spread out over the protein. If the sequence is rendered with no differentiation in how the symbols are rendered, it will be impossible to tell which are in the contact zone and which are not. See top figure in figure 3.3, here we have rendered all symbols with same parameters and you can not tell which are part of the contact zone. To differentiate between the symbols we have taken a straightforward approach in our software. The Amino Acids which are a part of the contact zone are emphasized by making them bold. While the important ones are bold, we want to make the non-important Amino Acid symbols less striking and this we do with making the symbols blend more with the background, where the important ones are a part of the contact zone, and the non-important are not a part of the contact zone. In the bottom figure in figure 3.3 you can see the where the important and non-important Amino Acids are placed.

If you look at figure 3.3, you can see that some symbols are rendered of-page. This figure is done with the intent to show without doing anything to the sequence length, you will not be able to see everything. In best case scenario, all the important Amino Acids are located in the same place on the sequence and you can see all of them at once, and in worst case they are spread out and depending on how big the protein is, you would need to shift the focus of the image around a great deal. When you move around its harder to get a picture of how everything is placed in regards to each other. Also as you can see in the bottom figure of figure 3.3, we can only see 5 important Amino Acids on the page while there are 19 important Amino Acids that are not visible on the page right now. In the sequence which we have used in the figures, there are a total of 216 symbols. In this sequence there are only 24 important Amino Acids, which means that 89% of the sequence consists of non-important Amino Acids. The Amino Acids which are not part of the contact zone are not always of interest. With the intensive to want to focus on the important Amino Acids and most of the space is used by non-important Amino Acids, we can reduce the size of the non-important ones. We do not want to remove them from the sequence, because we still want to keep an approximated context of how the important Amino Acids are placed in the protein chain, and to each other. A technique we can use to still keep the context we want, without removing the non-important Amino Acids is to abbreviate them.

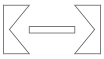


Figure 3.5: The abbreviation symbol

To abbreviate them we divide them up into chunks, where a consecutive sequence of non-important amino acids are set into one chunk and so on, as shown in figure 3.4. Then each chunk can be abbreviated, we use the abbreviation symbol shown in figure 3.5, where the shapes indicate start and end, and the line between them indicates how many symbols have been abbreviated. The longer the line, the more has been abbreviated.

We abbreviate instead of removing them, to still keep the context of where the important amino acids are in the sequence. When abbreviating just requires to merge them, a user might want to be able to enlarge the abbreviated chunk to see which Amino Acids are between the important Amino Acids, or to see how many there are. When we enlarge an abbreviated chunk, one can have an adaptive approach, instead of switching between showing nothing or showing everything in the abbreviated sequence. To do this adaptive expansion of the abbreviated sequence, enlarge the closest symbol to any important amino acid in its sequence until the screen space is filled up.

We add color on top of the symbols, the color represent which pairs it belongs to. This gives a quick view to see how many pairs a Amino Acid is a part of, and if you managed to get a unique color for each of the pairs, this will also give an indication of the Amino Acids that each of the Amino Acids in the contact zone is connected to.

Connection

We have gone through how we set up the sequences of the Selection tool, now we will connect the sequences to the graph based part of the Selection Tool. We want to have this graph to provide a visualization to show the interacting pairs, which pairs of amino acids are present in a given configuration and to be able to identify the differences in a set of configurations.
In our graph we represent each Amino Acid as a node and each Amino Acid pair as an edge. We want the graph to connect the two sequences, to give a visual representation of where in one protein chain an amino acid is connected to another amino acid in the other protein chain. To do this, we transform the Amino Acids symbols location from its place in the sequence to a 2D coordinate in the "canvas", this means we can interpret that the Amino Acid has a local position in the sequence it belongs to, and the global position of the symbol is in the whole Selection Tool area as shown in the figure 3.2 earlier. With the global position of the symbols, we can draw lines between the Amino Acids in the Contact Zone and this will give us the unbundled edge you see in figure 3.7.
With the straight edges between each Amino Acid pair, the graph will have obstructing edges. The figure we have presented has only 46 amino acid pairs, and it is reasonable to guess that the number of Amino Acid pairs in other configurations can be of a greater number. To try to avoid this obstruction, or at least try to save both screen space and keep some information of this graph, it is common to enhance the graph. There are multiple ways to enhance a graph. One can render the node with different size and color depending on chosen variables, one can use some layout for the nodes to spread them out achieve least amount of interference of the other nodes and overlap of edges. In our software we want to keep the sequence as it is. If we do not keep the sequence as is, the context of where a Amino Acid is in its protein can diminish. Therefor we cannot change the nodes position, size or color to give additional information. With this we are left with enhancing the edges. In our software we avoid adding to much extra information to the thickness of the edges, and we will explain later why we don't add color to all the edges at once. The technique we

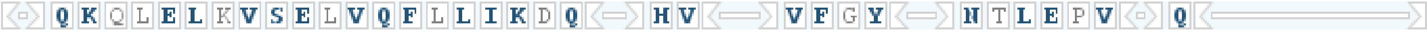

Q K Q L E L K V S E L V Q F L L I K D Q H V V F G Y N T L E P V Q

Figure 3.6: A abbreviated sequence

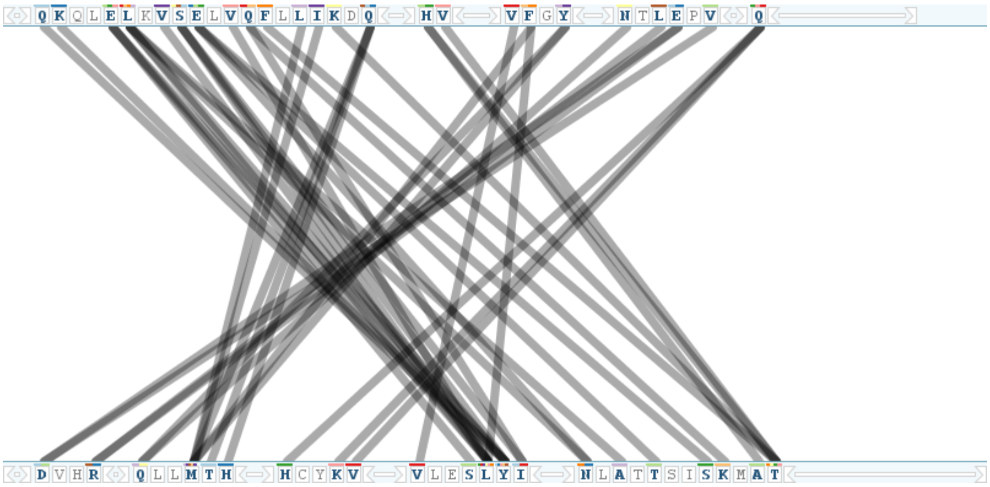

Q K Q L E L K V S E L V Q F L L I K D Q H V V F G Y N T L E P V Q
D V H R Q L L M T H H C Y K V V L E S L Y I N L A T T S I S K M A T

Figure 3.7: The graph without bundling.

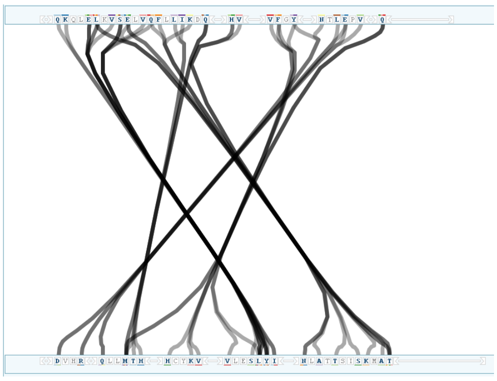

Q K Q L E L K V S E L V Q F L L I K D Q H V V F G Y N T L E P V Q
D V H R Q L L M T H H C Y K V V L E S L Y I N L A T T S I S K M A T

Figure 3.8: Bundled with Mingle edge bundling technique

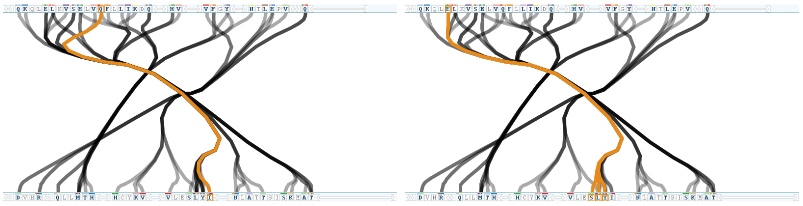
are left with for improving the graph is edge bundling. Edge bundling is a technique where you find edges of short distance to each other or some equal attributes, and you merge them together. A standard approach to edge bundling is to divide the edge up into extra temporary nodes and then iteratively going over each node and push them closer to other temporary nodes which are close to it. For our software we chose to use a edge bundling technique which is called Mingle and was proposed by Gansner et al [GHNS11]. The reason we chose it, was because it gave satisfactory result and is also a fast edge bundler. The high level idea of Mingle can be described as : "If you have multiple wires, and you want to bundle them together to get some structure in the wire mess. You start with picking a wire. While you hold it in your hand, you take the next and checks if it can be added to the wire in your hand and if it can, you also put this in your hand. After you have bundled all together ones, you check on the bundles if they can be bundled together with other bundles.", If you look at figure 3.8 you can see how it will look bundled.

With the edges bundled together like this, its hard to see where a single connection go, but you can see the overall picture of how the majority of the connections go forth and back between the two sequences. With this it is possible to identify differences in the configurations. With the connection set up, and enhanced as much as we can. We are going to explain the functionality of the graph, which will be used for both exploration in the graph and selection of pairs to be visualized in the 3D view. The functionality we have added to the Selection Tool is "Hovering" and "Selection". The hovering function is when you lets the mouse pointer hover over a edge or a symbol. When you hover over a edge, the edge will be rendered with an orange color and be shown on top of all the edge. With hovering over a symbol, all the pairs which this Amino Acid is a "member" of will be shown on top of all the edges and will be rendered in an orange color. The reason for hovering is to easily let the user look for a specific pair or how a Amino Acid is connected. The selection is for when you have found something interesting while using hover, when selecting a edge it will rendered with the color associated with the pair which is associated with the edge. It will also be rendered atop all the other unselected edges, so you will be able to easily see where it goes from and to. With the selection of a symbol, all the edges which it is a node in will be selected and again rendered with the specific color associated with the pairs which belong to the edges. If you see Figure 3.9 and 3.10 you can see how hovered and selected will look in the bundled graph.

3.3 3D Views

The Selection Tool which we now have covered is the basis for which Amino Acid pairs will be visualized in the 3D Views. To let the user see where a connections of the protein protein interaction is spatially positioned with regards to the proteins we have the Outline visualization for the pairs, with this we wish to be able to visualize the contact zone of the PPI. The second option for the 3D view, is an image in image visualization. This option will give us a visualization in the 3D view which is of one connection and is rendered beside the main visualization so that a user will be able to see a amino acid pair unobstructed. The reason we want this view is to be able to better see the amino acids spatial orientations to each other in the pair.

Figure 3.9: Left : Hovering over a single connection, Right : Hovering over a symbol



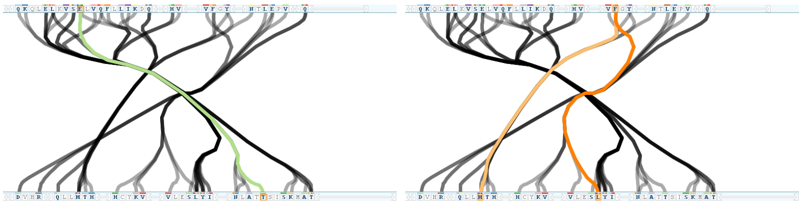


Figure 3.10: Left : Selected a single connection, Right : Selected a symbol

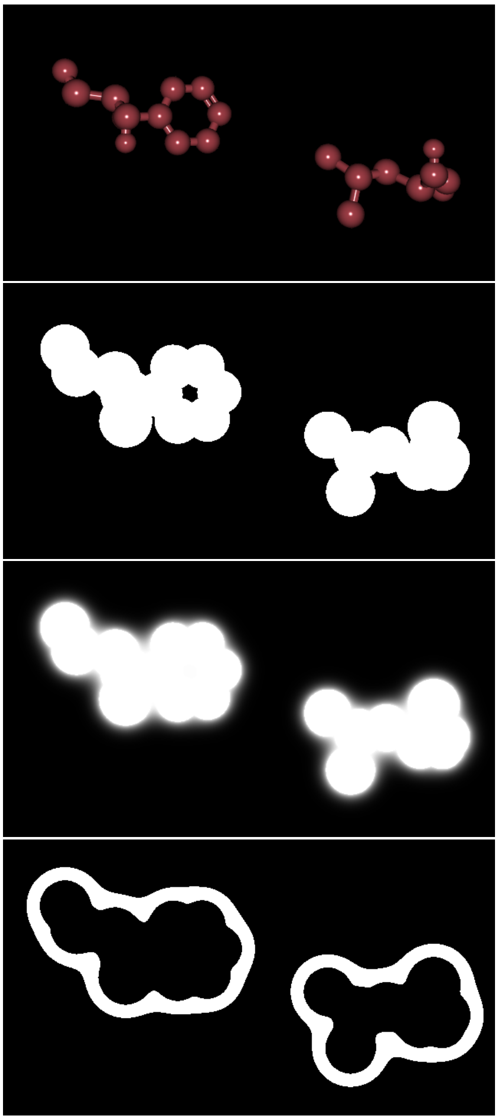


Figure 3.11: From top to bottom : Atoms to render, flat rendering, halo rendering, contour rendering

Outline visualization

To help with locating the Amino Acid pairs in the contact zone, which we want to further explore after the Selection Tool, our software provides a way to outline the pair. With the outlining of the pair(s), they are easily distinguishable from the rest of the PPI. This lets us explore the contact zone, arguably only one at a time, with seeing how the pairs are positioned in the PPI and can distinguish if they are far or close together. The straightforward process to indicate where a pair is in the 3D view, would be to draw a circle around the whole pair as seen in Figure 3.12. From the illustration you can see that the circle will mostly hold empty space. We wanted to improve on this, to have a more snuggly fitting outline we took inspiration from Collins et al. proposed Bubble Sets[CPC09]. The Bubble Sets with the use of iso contour lines and marching cubes, provides a continuous bounding contour. This

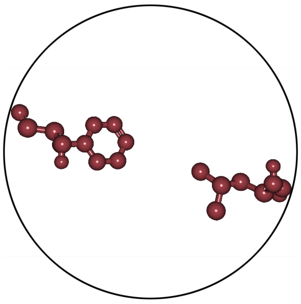


Figure 3.12: Amino Acid Pair with a circle outline.

continuous borders is something we would like to have, as it encapsulate the Amino Acids better than the circles, and we are left with less empty space in the outline around the Amino Acids. In our software we don't use the same approach as they did to achieve the outline result. We added a glow around the Amino Acids, and traced the outlines of the glow to get the contour. We will further on explain the different steps we took to achieve the outline result.

In this thesis we have mentioned multiple times that the part which will be rendered in 3D view is decided by what you select, the reason we do not render all the Amino Acid pairs in the contact zone in our 3D view, is because as seen Figure 3.13 they would overlap each other and cause obscuration of each other. Making it hard to discern any information, regarding any specific Amino Acid pair from the contact zone. This is why we have focused to rather let the user chose which specific Amino Acid pairs they want to look at in the 3D view rather than visualization the whole set of amino acids from the contact zone.

To get our contour around the Amino Acid pair, we have to render "flat" to get seeding image for our glow, and then we can render our contour from the glow. See Figure 3.11 In our software we use more an approximated flat shading of the Amino Acid pair, as we do not render a pair and then render it black and white depending on if there were something there or not. We take each of the atoms in the Amino Acid pair, and around their coordinates we render a flat circle pointing toward the screen. The size of the circle is user dependent, this can help later if you wish to have a bigger area for your contour or glow.
For our glow around the Amino Acids, we chose to use a blur around the flat shading.

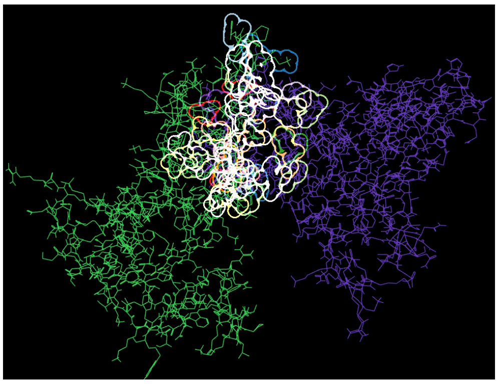
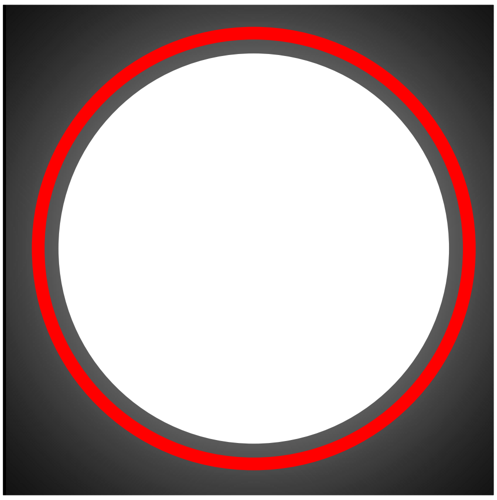


Figure 3.13: All the Amino Acid pairs in a configuration is rendered at the same time.



r

Figure 3.14: The white is the original flat image, the glow is after a halo rendering, and the red circle indicates where we draw our contour based on the "power" left from the halo.

We chose to do a Halo shading as proposed by Bruckner et al.[BG07], we chose the halo as it would not distort the original image, and give us a nice glow. Halos are something used by illustrators by taking use of the humans ability to discern quite well small contrast by drawing halos around the object. They can be dark borders to give some depth, or glowing around the object to detach them from the rest. We want to use the glow to detach the object from the rest. We let the user stop the contour process stop here, if it is of more interest to see the Amino Acid pair detached by glow, rather than a contour around the pair.

Contour is used to show only the outlines of a specified target. In our software we use the Halo rendered texture as a "seeding" image, and based on the "power" from the halo, if the value from the halo is between a range specified by user.Greater range will give a thicker contour. draw in the border there, Figure 3.14 red circle shows how we render the contour of the Amino Acid from the glow. With this contour, if the Amino Acids in the pair are not close together in the screen-space, they will not share a contour. This will give an indication on the closeness of the Amino Acid pair as well as the ability to locate their position.

With this process, each of the pairs need to be rendered separately. Since we do not want the information from each of the pairs to be shared by each other. If the spread of all the Amino Acid pairs selected are shared, you will get a continuous contour around all the Amino Acid pairs instead of a contour around each of the pairs. The result of not rendering them separately will look like it does in Figure 3.15

Image in Image

To better achieve information on the spatial orientation of the Amino Acid pairs to each other, we implemented the image in image visualization. With this view we will get a unobstructed rendering of a selected Amino Acid pair, and we will be able to tell how they are rotated to each other and a general idea if they are close or far apart. The approach to achieve this visualization is done with selecting a Amino Acid pair.

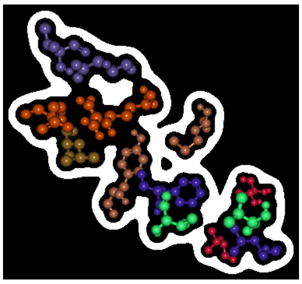


Figure 3.15: All Amino Acid pairs rendered together, with letting the information merge between all.

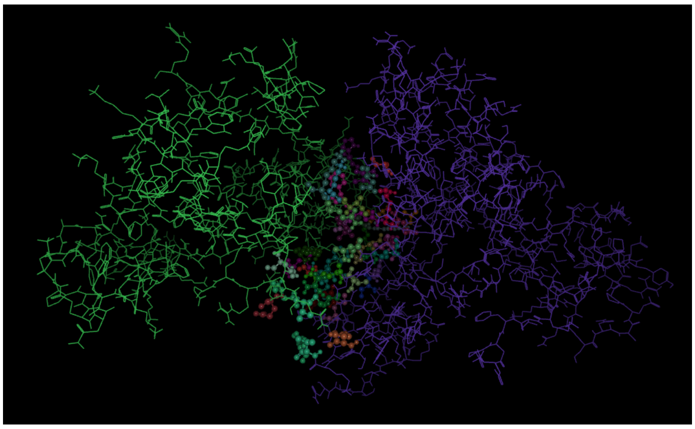


Figure 3.16: Contact Zone rendered as Ball and Sphere, rest is rendered as Wireframe.

Render the selected Amino Acid pair with a molecular visualization of choice, to its own texture. Finally add this texture to the main rendering window, the way to do is to chose a layout pattern. In our case we did it simple, and chose 8 static places on the border of the screen. And render the texture to its appointed space in the main rendering window according to a pattern of choice.

Selection option

In our process when we used Cavers software, to get the Image in Image view we used their Selection rendering methods. To achieve this we had to create some Selections which we could later render. But they were by default visible, so we ended up having all Amino Acid pairs rendered as selection on top of the original Molecular representation of the PPI. This turned out to work quite well, as it gave a additional unobtrusive view of where the Amino Acids were in the PPI. So this view was stumbled upon, but to achieve this, you render the PPI in a chosen manner of non surface rendering of the Molecular Visualization, and then you render the Amino Acids pair in a different molecular visualization to achieve a non obtrusive manner to show where all of them are in the view. This can help with locating how the contact zones look like. When it comes to coloring this, you can either chose to have them all rendered in the same color to avoid giving information and then render them with the same color of a selection if something is chosen.

Chapter 4

Implementation

In this chapter we will go through how we implemented our software. We implemented our software as a module for Caver Analyst, specifically the Caver Analyst 2.0 Beta version. Everything is written using Java and utilizing OpenGL 2.0. The data we use in our software is generated by the use of HADDOCK[DBB03] which gives us the data for a number of configurations of a protein-protein interaction, in separate contact files, which contains the information on each of the connections which are in the loaded configuration. We use generated configurations of nse1-nse3 Protein-Protein Interaction with only cluster 1 in our figures.

4.1 Caver Analyst

Caver [Kvv$^+$] is a software tool for visualizing and analyzing tunnels and channels in protein structures. Caver provides all the molecular visualization shown in this thesis. We have only implemented the high level specifically for Protein Protein Interaction.

4.2 Selection Tool

In selection tool we are going to cover how we have implemented the adaptive expansion of abbreviated chunks in the sequences, and how we edge bundled in the graph chapter of the selection tool.

Sequence

We take into account that a user would like to see which Amino Acids are in between the Contact Zone, but would still like to keep everything on the screen. So we have implemented an adaptive enlargement of the abbreviated chunks. A chunk is in our case, a number of consecutive amino acids which are not in the contact zone with a set larger than two consecutive amino acids, as shown in Figure 3.4. The reason we do not abbreviate 1-2 amino acids are because the symbol we use for our abbreviation takes up more space than two symbols. A chunk will consist of multiple segments, a

Algorithm 1 Adaptive Enlargement

```
if chunk is abbreviated then
    while Screenspace left do
        Expand closest segment in chunk
```

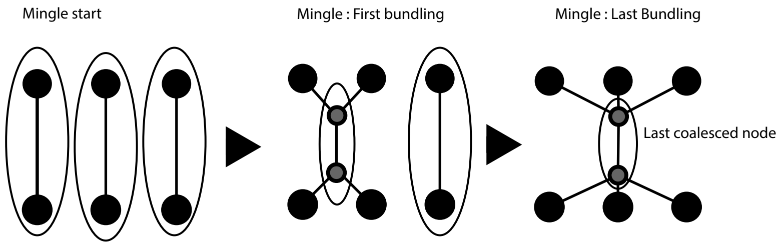


Figure 4.1: Steps in Mingle algorithm.

segment will contain all amino acids that share a range of the same closest distance to any interacting amino acids in the same sequence. Our range is set to be 1 Ångström in difference, i.e everything in range [1-2] Ångström away from their closest interacting Amino Acids are in the same segment.

Algorithm 2 Chunk segmentation

```
for All Amino Acids in Chunk do
    Add to Min Priority Queue
while Min Priority Queue(MPQ) is not empty do
    if Segment is empty then
        Pop MPQ and add to Segment
    else
        Peek MPQ
        if Peeked elements Distance is in range of segment then
            Pop MPQ and add to Segment
        else
            Create new segment in chunk
```

As we show in our Algorithm 1, after we have segmented each chunk. When a user wishes to expand on the abbreviated chunk, we expand each segment until we reach the limit of the screen. This will utilize the screen, so you can still see as much of the sequence on the screen as possible.

Connection view

For the graph chapter of the selection tool we will explain our edge bundling and how we draw the lines. Our Edge Bundling technique is Mingle[GHNS11] as mentioned in methodology. We used the implementation done by Belmonte [Bel15]. He implemented a JavaScript version of Mingle, and in our software we translated it from JavaScript to Java. The way Mingle will give us the nodes in the end were not sufficient for our use, as we needed to have access to each single path for each of the edges to be able to distinguish between them. After Mingle is run you will get back the last coalesched nodes which are the parent nodes for all paths, as seen in figure 4.1. To get the path for each of the edges we used a recursive algorithm.

We looked at all the nodes left in the Mingle Graph, which after running the Mingle algorithm should be left with the "top" bundled nodes. We go through each node, and for each node we look at all its children, and the children of the children and so forth until we reach a leaf, i.e a node that has no children. We gather up the coordinates for each node down to the leaf, and when we get to the leaf which is the original coordinates to the Amino Acid pair, we give the path to the corresponding Amino Acid Pair. The reason we want all connections to have their own path, is so that they can later be selected individually.

4.3 3D View

This chapter will focus on the Outline visualization of a relationship and the Image in Image visualization for a single connection. the 3D view has utilized OpenGl 2.0 and its shaders are written in GL Shading Language (GLSL). All except the Flat Shading are just using the Vertex shading and Fragment shading, for Flat Shading we also added a Geometry Shader to the rendering pipeline.

Relationship outliner

As we mentioned in Methods chapter for the rendering of connections, to avoid the data from them from mering together in the rendering, we have to render them separately. To avoid going through the rendering pipeline for each selected amino acid pair, we divide them up into channels and textures, for texture we use 2DLayeredTexture. As seen in Algorithm 3, we put as many who will not interfere with each other in the same channel. The channels we use are R,G,B and A, and if we have more than four groups active we add another layer to the texture, and start with R,G,B and A again. To get which Channel a group of amino acid pairs belong to, we use this method : $Channel$ = group number%4, and : Texture Layer = group number/4 to see witch layer needs to be active in the rendering.

Algorithm 3 Group division

```
for All Selected Connections do
    Find BoundingBox in viewspace
while Not all Selected Connections have a group do
    Find all Selection with non overlapping Bounding Boxes
    Add them to same group
for All groups do
    Give each group its channel
```

Further to achieve our outline rendering we need a seeding image for the contour, which is the Halo rendering and for the Halo Rendering we need another seeding image which is our Flat rendering. Therefore we will start with going through the Flat Rendering, then Halo and finally contour.

The Flat rendering goes through all selected amino acid pairs, and for each amino acid pair it will load all atom positions, of the atoms in the amino acid pair, into the rendering pipeline. After this it will load which channel it belongs to and which texture layer the connections should be written to. In the geometry shader it will create a square around each atoms position which faces the camera. In the Fragment Shader it will be rendered based on how close a fragment is to the center of the atoms position. This will leave us with an approximated flat rendering of the connections.The Halo we used for this is a Halo rendering made for 3D volume proposed by Bruckner et Al.[BG07]. As they propose they use a seeding image generated from a volume image, but in our rendering we use the Flat rendering. The rest of our Halo rendering is the same approach Bruckner et Al[BG07] uses, except that we run it once for each layer in the layered texture. The reason we do not need to run it once for each active group, is because when you add together colors in the rendering they do not affect each other.

When we now have the texture after the Halo rendering, we use this data for the Contour rendering. For each layer we run the contour shader, and in the shader we look through all the channels in the location of the fragment we are looking at. If a channel is between a range of 0.1 and 0.9 in "power" we set the respective channel to 1.0 or else its just 0.0, see Algorithm 4. This gives us a "blobby" countour around our Amino Acid pairs after the glow from the Halo shading.

Algorithm 4 Contour Shading

```
2DTextureSampler HaloTexture
Layer
$vec4$    $Channels = texture(HaloTexture, vec3(frag.xy, Layer))$
Find BoundingBox in viewspace
for Channel : Channels do
    if Channel $>$ 0.1 || Channel $<$ 0.9 then
        OutColor[Channel] = 1.0
    else
        OutColor[Channel] = 0.0
```

Image in Image

In our implementation of Image in Image visualization, we relied heavily on functionality from Caver. When we created the data we created a structure selection of each of the connections. A Structure Selection in caver consist of making a smaller set out of the atoms already available, which you would further like to look at. As you can see from a figure in our program, when we don't outline any connections, and there is still some selections rendered in another fashion than the rest of the atoms. This is due to when we utilized Caver's Selection functionality, and not being able to turn them invisible later. This leaves the Selections rendered in another fashion and color than the rest of the atoms in the proteins.

The reason we wanted to use the Selection functionality from Caver, is since it can be rendered separately in another molecular visualization than the rest of the structure. This gives us the possibility to run the rendering of this separate selection. By using our own Frame buffer object, instead of the default one which renders to the screen, we can render it to our own texture. After we have rendered it to our own texture, we can then render it to a selected location in the main window, and as stated in methodology we have 8 static placements around the border of the screen.

Chapter 5

Results

In the Methodology chapter we mentioned some questions, and those questions were:

- Q1 : Which configurations contain a selected interacting pair of amino acids and what is the frequency of occurrence of this pair in all configurations?

- Q2 : Which pairs of amino acids are present in a given configuration?

- Q3 : How close are the amino acids in the contact zone and which are the closest ones?

- Q4 : How similar and different are the contact zones in the configurations?

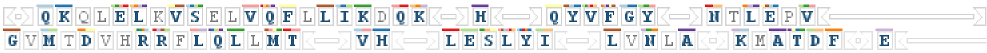
- Q5 : What are the differences between the sets of amino acids in the contact zones of configurations?

We will look at our softwares capability and answer if a user could explore and answer these 5 questions.

Our software provides the possibility to see if a configuration contains a selected interacting pair of amino acids, and you can also find the frequency of these. But this this would have to be done manually by the user. As you can see in the figure 5.1, with the color given you can see that the Amino Acids Q-M are most likely a pair. As we mentioned earlier, we don't have a separate color for each of the Amino Acid pairs, we use 12 different colors which will repeat if there are more than 12 Amino Acid pairs in the contact zone, and therefor you will get a probability that they are a pair. To be certain you will have to either hover or select the connection to check as can be seen in figure 5.2. In the figure we show how you can see that it is possible to find a selected pair of Amino Acids and the frequency, in this case the pair was two Q-T pairs. With the edge bundling we have utilized it is hard to see a single connection, so the easiest way to search is to look for one of the Amino Acid symbols and get all the connections it is involved with. The connection view in Selection Tools will show you how many of the Amino Acids in each of the proteins are in the Contact Zone. A user will be able to determine if there are many connections depending on how dark and how many edges there are, it will be possible to determine if the Amino Acids in the contact zones are evenly spread out in the protein or if they are all close together.

Figure 5.1: Sequences to the proteins with interaction from a configuration

(a) Sequence of Protein A


Q K Q L E L K V S E L V Q F L L I K D Q K H Q Y V F G Y N T L E P V
G V M T D V H R R F L Q L L M T V H L E S L Y I L V N L A K M A T D F E

(b) Sequence of Protein B

In out module to Caver, we have not implemented any specific way to search for the distances between the Amino Acids in the contact Zone. We have a function which lets you select the pair with the least distance or the greatest distance between them. Which were used in figure 5.4 to get the pairs we have visualized. Caver have implemented a measurement tool which lets you see the distance between two atoms, and with the use of our Outline Visualization of a pair, you can locate and select two atoms, one from each Amino Acid in the pair, to see the distance between them as we have done in figure 5.3. This shows that a user can explore the distances between Amino Acid pairs. With the help of the Image in Image visualization option, it is also possible for a user to determine the closeness of a pair to each other and the spatial orientation to each other as shown in Figure 5.6.

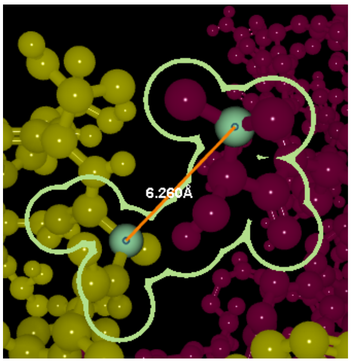


Figure 5.3: Distance between the closest Amino Acid pair in the configuration. The distance is sorted by the Alpha Carbon position, which is not necessary the atoms selected in this view.

With the Selection Tools sequence and the connection between them, a user might not be able to identify a specific Amino Acid pair at a glance. But a user will be able to see similarities and differences in the contact zones of different configurations. We have the figure 5.5 where we have the Selection Tool presented from 4 different configurations of the nse1-nse3 complex. We can see that there are similarities between Configuration 5 and 21, they are both twisted, i.e the Amino Acids in the beginning of Sequence of Protein A connects to Amino Acids in the end of sequence of Protein B. With Configuration 148 and 200, we can see they they both are less twisted than 5 and 21, as they have more edges going from the beginning of Sequence A to the beginning of sequence B. We can also tell that the Contact zone between configuration

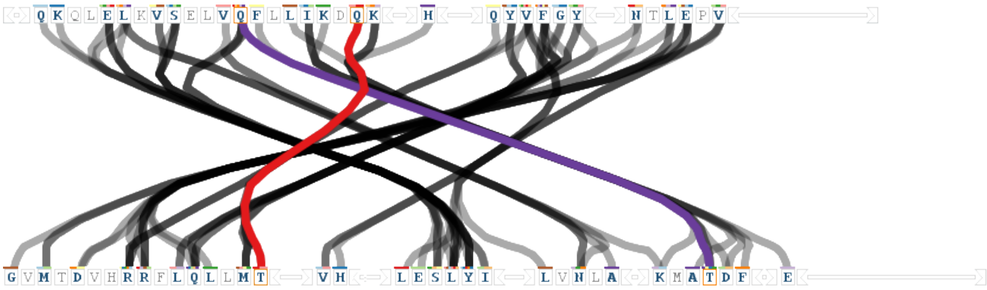

Q K Q L E L K V S E L V Q F L L I K D Q K H Q Y V F G Y N T L E P V
G V M T D V H R R F L Q L L M T V H L E S L Y I L V N L A K M A T D F E

Figure 5.2: The sequences with the two pairs of Q-T selected

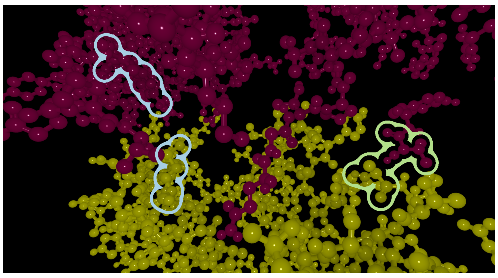
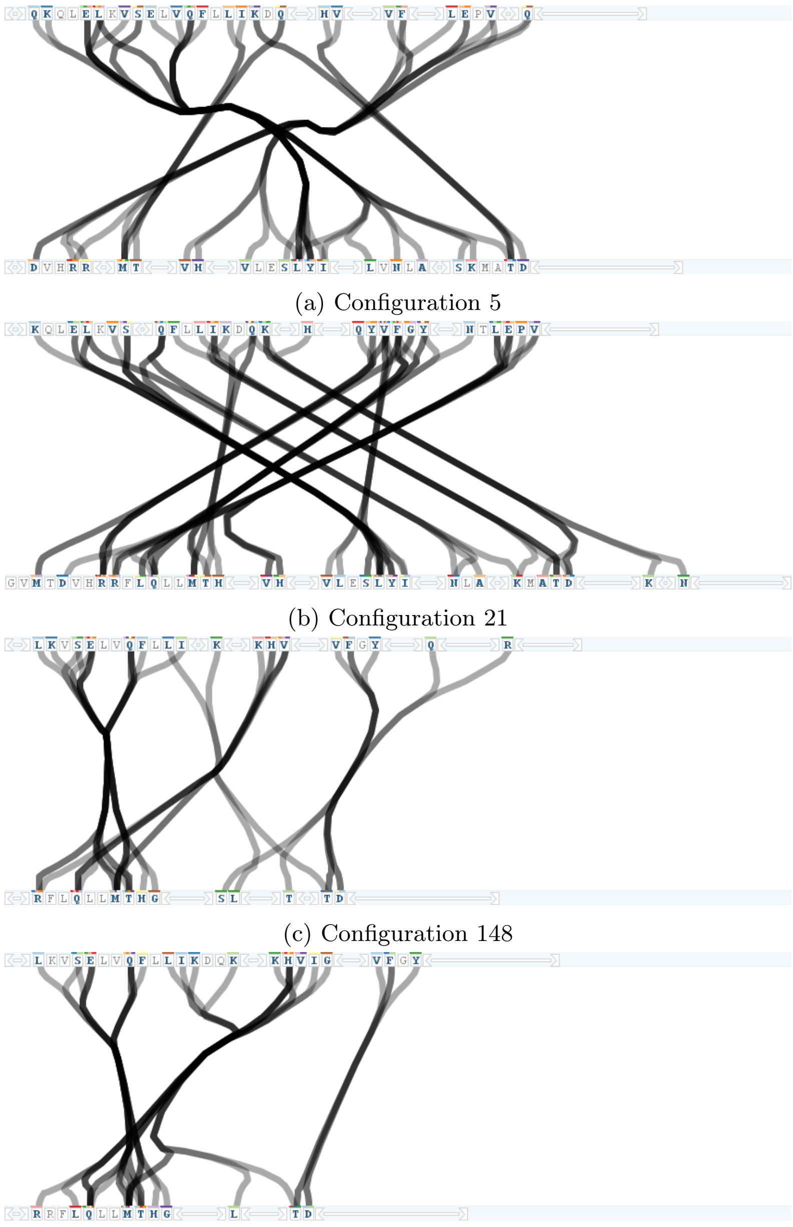


Figure 5.4: Blue contour indicates the pair which has the greatest distance between them, the green contour indicates the closest pair in the configuration.

148 and 200 are quite similar, where the majority of the Amino Acids in the contact zone are the same, this does not tell us explicitly that the connections are the same. With this we can also tell that Configuration 21 is the Configuration which is has the most Amino Acid pairs in its contact zone, and the contact zone which is most spread out over its protein sequences. By the color atop the symbols in the Selection Tool you can also tell which Amino Acid is the most connected, and it is possible to see that the Amino Acid which are in most pairs, differ from each configuration.


Q K Q L E L K V S E L V Q F L L I K D Q H V V F L E P V Q
D V H R R M T V H V L E S L Y I L V N L A S K M A T D
K Q L E L K V S Q F L L I K D Q K H Q Y V F G Y N T L E P V
G V M T D V H R R F L Q L L M T H V H V L E S L Y I N L A K M A T D K N
L K V S E L V Q F L L I K K H V V F G Y Q R
R F L Q L L M T H G S L T T D
L K V S E L V Q F L L I K D Q K K H V I G V F G Y
R P F L Q L L M T H G L T D

(a) Configuration 5

(b) Configuration 21

(c) Configuration 148

(d) Configuration 200

Figure 5.5: The connection of 4 different configurations.

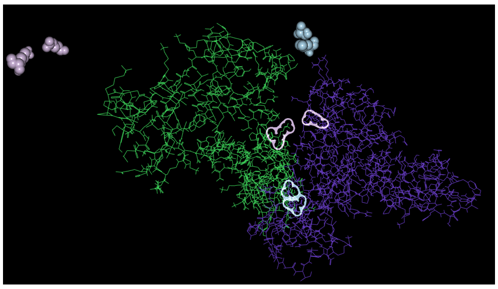


Figure 5.6: Two amino acid pair shown in the Protein-Protein Interaction, and rendered beside the PPI.

Chapter 6

Discussion and Limitations

A large limitation is that this has been done without inputs from any domain experts. As we mentioned in Methodology, a limitation in our software is the ability to visualize multiple Amino Acid pairs at once. As we render them separately, runtime will decrease if to many are selected at once. Realistically the amount of Channel groups we can have, will not be the problem so we can accommodate for large selections of Amino Acids pairs if one do not take into account the delay or the occlusion caused by the amount of selected pairs.
Another limitation is the amount of unique colors that are required, we use 12 unique colors from ColorBrewer. ColorBrewer is a website which contains multiple color schemes meant to provide colors which provides as much information as possible. Because we only use 12 unique colors, and there are most of the time more than 12 different Amino Acid pairs in the contact zone, we will have repeating colors. This means that we can not depend solely on the colors to locate a specific Amino Acid pair.
In our software we focused on screenspace saving edge bundling, this means that we have reduced the number of pairs you can see at a glance in the Selection Tools connection view. It was designed with the intent that there would be multiple configurations, and the edge bundling would contain as much information to where the overall pair was connected, i.e if the PPI contact zone was twisted as shown in the result chapter, or how many connections there are.
The Image in Image visualization is limited to 8 specific locations around the border, this is something which could be improved with inspiration from Labeling algorithms.

In our software the selection which shows the contact zone was made by accident, and not properly fixed to accommodate the functionality it could hold. With correct coloring of the selections it would provide a clear view of which selection were selected, but because they were made as multiple connected selections, we could not get to render the specific Amino Acids in the correct color. The problem was also that they were multiple Selections, laying a top each other, meaning that by changing the color of a selection belonging to a specific amino acid pair, did not mean this would be the one laying at the top which is the one we would see.
The Flat rendering could have taken an extra step, where we could first the same approach as we did to get the texture for the image in image visualization, and get an accurate outline of the atoms. This could be both good and bad, depending on which rendering type you used and what you wanted. i.e if you used the line rendering of

a molecular visualization the contour would be smaller and the pairs would be less connected by the contour and would rely more on color. But our Halo rendering would still give the select amino acid pair(s) a detached look helping picking them out.

The image in image visualization in our software is limited to 8 extra being visualized, this is because we had the simple layout of having the specific location around the border. This visualization could be changed, as the main thing needed to obtain from this view was the spatial orientation of the amino acids in a amino acid pair, to each other.

Chapter 7

Conclusion and Future Work

With our software, a domain expert would have been able to answer all the questions which we stated both in the methodology and result chapter. Some of the questions would take time to answer, and there would be definitely possibilities to improve on these. One of the things that could be added would have been a search tab, to specifically search for a Amino Acid pair and select all the relevant pairs from the contact zone.

We had some feedback on both the Connection View of the Selection Tool, and the Selection option in the 3D view. The domain experts wished it would be easier to see each connection, and have an edge bundling technique which would resemble more a subway line illustration, i.e the edges drawn would be restricted to be close to 45 or 90 degrees between sub nodes as created from Mingle. This means the edge bundling in future would be improved with a effort to show more connections. And the Selection option has the problem which we mentioned in Limitations, this would be remade so we could change the color of each single Amino Acid and show the correct color which the selected pair represented.

The Image in Image visualizations layout would be changed, to something inspired from labels. Since this view can give us an idea of the orientation of the amino acids to each other, it does what it was intended to. But it is able to accommodate to few pairs at once.

With regards to unique colors, there are papers on generating and we would improve on more unique colors. As we mentioned in Limitations with our Image in Image visualization, this would be improved with inspiration to Labeling algorithms to be able to have more than 8 at once.

It is also worth mentioning, physiochemical properties are something domain experts are also interested in and this is something which could be added. The possibility to visualize the Amino Acids physiochemical properties, and the pairs properties.

There are room for improvement with this software, but as far as answering the questions go, this software can be of help.

Bibliography

[AGC13] Giuseppe Agapito, Pietro Hiram Guzzi, and Mario Cannataro. Visualization of protein interaction networks: problems and solutions. *BMC bioinformatics*, 14 Suppl 1(SUPPL.1):S1, 2013.

[Bel15] Nicolas garcia Belmonte. Multilevel agglomerative edge bundling in javascript. https://github.com/philogb/mingle, 2015.

[BER04] Yih-en Andrew Ban, Herbert Edelsbrunner, and Johannes Rudolph. Interface surfaces for protein-protein complexes. 2004.

[BG07] Stefan Bruckner and Eduard Gröller. Enhancing depth-perception with flexible volumetric halos. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1344–1351, 2007.

[CPC09] Christopher Collins, Gerald Penn, and Sheelagh Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009.

[DBB03] Cyril Dominguez, Rolf Boelens, and Alexandre M. J. J. Bonvin. Haddock: a protein−protein docking approach based on biochemical or biophysical information. *Journal of the American Chemical Society*, 125(7):1731–1737, 2003. PMID: 12580598.

[FBG$^+$] Katarína Furmanová, Jan Byská, Eduard M. Gröller, Ivan Viola, Jan J Palecek, and Barbora Kozlíková. COZOID : Contact Zone Identifier for Protein-Protein Interaction Analysis.

[GHNS11] Emden R Gansner, Yifan Hu, Stephen North, and Carlos Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 187–194. IEEE, 2011.

[JJQH15] Fleur Jeanquartier, Claire Jean-Quartier, and Andreas Holzinger. Integrated web visualizations for protein-protein interaction databases. *BMC bioinformatics*, 16(JUNE):195, 2015.

[KBE09] M. Krone, K. Bidmon, and T. Ertl. Interactive Visualization of Molecular Surface Dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1391–1398, 2009.

[KBJM12] Martin Krzywinski, Inanc Birol, Steven JM Jones, and Marco A. Marra. Hive plots-rational approach to visualizing networks. *Briefings in Bioinformatics*, 13(5):627–644, 2012.

[KKF$^+$16] B. Kozlíková, M. Krone, M. Falk, N. Lindow, M. Baaden, D. Baum, I. Viola, J. Parulek, and H.-C. Hege. Visualization of Biomolecular Structures: State of the Art Revisited. *Computer Graphics Forum*, 00(00):1–27, 2016.

[Kvv$^+$] Barbora Kozl'ikov'a, Eva vSebestov'a, Vil'em vSustr, Jan Brezovsk'y, Ondvrej Strnad, Luk'avs Daniel, David Bedn'avr, Anton'in Pavelka, Martin Mavn'ak, Martin Bezdveka, Petr Benevs, Mat'uvs Kotry, Artur Wiktor Gora, Jivr'i Damborsk'y, and Jivr'i Sochor. CAVER Analyst 1.0: Graphic tool for interactive visualization and analysis of tunnels and channels in protein structures. *Bioinformatics*, 30.

[LR71] Byungkook Lee and Frederic M Richards. The interpretation of protein structures: estimation of static accessibility. *Journal of molecular biology*, 55(3):379IN3–400IN4, 1971.

[LS11] Roman A Laskowski and Mark B Swindells. LigPlot + : Multiple Ligand À Protein Interaction Diagrams for Drug Discovery. pages 2778–2786, 2011.

[LVKW14] Marc F Lensink, Sameer Velankar, Andriy Kryshtafovych, and Shoshana J Wodak. CAPRI Round 30. 11, 2014.

[SA14] Tuba Sevimoglu and Kazim Yalcin Arga. The role of protein interaction networks in systems biomedicine. *Computational and Structural Biotechnology Journal*, 11(18):22–27, 2014.

[SOR$^+$11] Michael E. Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng Liang Wang, and Trey Ideker. Cytoscape 2.8: New features for data integration and network visualization. *Bioinformatics*, 27(3):431–432, 2011.

[STDB16] Joris Sansen, Patricia Thebault, Isabelle Dutour, and Romain Bourqui. Visualization of sRNA-mRNA Interaction Predictions. *2016 20th International Conference Information Visualisation (IV)*, pages 342–347, 2016.

[SXB07] Rohit Singh, Jinbo Xu, and Bonnie Berger. LNBI 4453 - Pairwise Global Alignment of Protein Interaction Networks by Matching Neighborhood Topology. pages 16–31, 2007.