FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA

MOLECULAR VISUALIZATION

FOONG YEE PENG
(WEK 020054)

Under the supervision of Ms. Mangalam Sankupellay
Moderator : Prof. Ir. Dr. N. Selvanathan

SESSION 2004/2005

A Graduation Exercise submitted to the Faculty of Computer Science and Information Technology, University of Malaya
In partial fulfillment of the requirement for the Bachelor of Computer Science Degree.

ABSTRACT

There are many programs for molecular visualization such as RasMol, Quanta, GRASP, SETOR, Amira, etc. They all perform many of the same basic tasks; they read in molecular structure data, perform some analysis, and display the results. Visualization programs usually analyze the data to display some aspects of it, while programs for analysis often return their results as detailed textual information. As visualization and analysis tasks become more complex and the expectations for programs continues to rise, it has become crucial to build a powerful molecular visualization system that can speed up the researches in chemical, pharmaceutical and medical areas. Thus, the purpose of this project is to discover a system that enables segmenting, rendering and visualization of static and dynamic molecular data at an appropriate speed.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API – *Application Programming Interface.*

DNA – *Deoxyribonucleic acid.*

ROI – *Region of Interest.*

MDL – *Molecular Design Limited.*

PDB – *Protein Data Bank.*

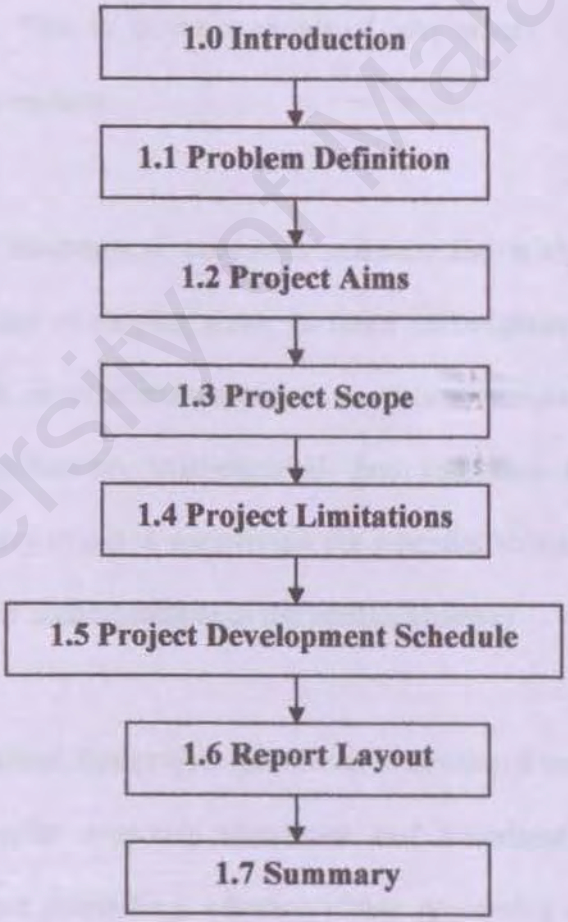
VR – *Virtual Reality.*

ZIB – *Zuse Institute Berlin.*

CHAPTER 1

INTRODUCTION

This introductory chapter gives a description or purpose of the project and problems to be solved. This part will discuss the significance and rationale of the project. Besides, it will also outline the system functions, limitations, and its assumption.



1.0 Introduction

The world is full of information. Simulations, experiments and data collections comprise of an enormous and permanently increasing accumulation of data. New ways have to be found to reveal the information hidden in huge data sets.

Visualization offers a solution at presenting complex information in a comprehensive way by exploiting the sensory apparatus and the highly developed perceptual capabilities of humans. Visualization denotes the activity of obtaining a visual representation of data. This is contrast to visual arts where the emphasis is on visualization of spiritual matters.

The understanding of biochemical processes requires the study of structure and dynamics of biomolecules of varying sizes. In these investigations, vast amounts of data are generated both in experiments and in numerical simulations. An interplay between chemical classification, mathematical data reduction and graphical data representation is necessary to aid in uncovering the essential structure of biomolecular process and making them understandable to the human observer.

In order to accomplish these, this project has aimed to develop a software environment which integrates molecular dynamics simulation and visualization. By combining current high performance computing, advance image processing and high rendering capabilities, such system will facilitate major progress towards molecular visualization.

1.1 Problem Definition

i. Diseases exist and have been a norm in our everyday life. While many are spread through air and interaction, however some are inherited through DNA. Most scientists don't really have a deep understanding on molecule which has become an obstacle to determine the solution for certain diseases.

ii. Studies on complex molecules have been refrained without an interactive display of the molecular data. Student are still coping with the two-dimensional display of DNA molecule represented by diagram consisting of A,C,T, and G with each representing the four nucleotides that compose a strand of DNA. Thus, many of them do not have the opportunity to gain insight by an interactive exploration of the structure and mechanical properties of molecular systems.

iii. Lacking tools which are able to read data from various databases. Without the integration of data from various databases, pharmaceutical researchers are unable to achieve their goals, identification of gene functions and getting a new drug for certain diseases. This is mainly because they can hardly determine the exact location of genes on chromosomes, lines or associations with disease as well as tissue information. Short to say, researchers in pharmaceutical field are lack of sense on how specific protein molecules behave.

iv. Molecular analysis takes a period of years without three-dimensional data sets for fast and powerful analysis. Structures and measurement obtained are mainly inaccurate without a wide range of analysis tools. Thus, this has caused the financial

cost to increase due to research that takes several years before finally come up with a solution.

v. Plant breeders and seed industries are suffering major lost due to weak genes in their crops. A tool is needed in the biotechnology field in order to identify important genes, thus to pinpoint specific traits and accurately identify seed varieties through DNA testing.

1.2 Project Aims

The goals of the research are as follow:

i. **Three-Dimensional Surface and Volume Rendering** – This research served its purpose in creating both surface and volume models of molecules.

ii. **Develop Techniques that Filter and Extract Essential Features** – Also known as segmenting, the main aim of this research is to segment static and dynamic molecular structure for the use in medical, biology, chemistry, or engineering field.

iii. **Visualize Metastable Conformations** – In visualization, one of the aims is to visualize metastable conformations in such a way that one is able to understand the differences between the various shapes a molecule may adopt. Here, in this research, it presents the methods for fast computation of probability density and depicting them in standard technique.

iv. **Research Tool** –This emerging research field is promising in order to improve the understanding of basic biological processes, leading to improvements in the diagnosis, treatment, and prevention of many genetic human diseases. This will thus reduce financial and time cost as well as to ease work burden.

v. **Study of Molecular Mechanism** – The central goal of molecular biology is to elucidate the relationship between sequence, structure, properties and functions of biomolecules. Such knowledge allows one to understand the molecular mechanism underlying biological or pharmaceutical processes. The two major steps in drug discovery are finding a chemical compound which shows a desired bioactivity and then modifying this drug candidate to build in all desirable properties.

All the aims stated here have placed the development of visualization in becoming a crucial element. Thus, it has brought to the research of molecular visualization to be carried out in this project.

1.3 Project Scope

The fields of users that will benefit from this system are as follows:

i. Pharmaceutical area – User in this field can gain insight about which genes to target for drug intervention, which choices might be dead ends, and which choices might be experiences for a team's quest to cure or treat diseases.

ii. Medical area – Medical staffs need not got through the hassle and stress as molecular visualization has ease the medical diagnosis process by providing a three-dimensional data sets for fast analysis. Early treatment and prevention can help to cure a patient illness, thus improving the life quality.

iii. Biotechnology industry – Deep understanding of DNA on certain group of crops can help increase the crop production with the least investment.

iv. Chemistry field – Chemist can both see and feel how the protein 'unfolds' when several hydrogen bonds in the protein are broken. Such tools and methods will provide all the necessities for many bioinformatics applications since researchers have completed the draft map of the human genome. With a tool that can visualize molecule, it has ease the process of discovering the structures and functions of all of the proteins encoded in it.

1.4 Project Limitations

The constraints which have limited the scope of the research in this project are as follow:

i. The visualization will only involves a specific molecular data, in this case, it emphasizes on DNA and alkane (trajectories) molecules. The file format are limited to PDB format (protein database), and .zmf format (trajectories).

ii. The visualization will only use the tools in AmiraMol 3.1 version, while some powerful tools might contained in other existing molecular visualization tools.

1.5 Project Development Schedule.

Molecular Visualization project started in June 2004. It is divided into a few stages, which are:

i. Planning stage.

ii. Literature Review.

iii. Methodology/ Technique Decisions.

iv. System Analysis and Design.

v. System Development and Implementation.

vi. Documentation.

The Gantt chart shown on next page depicts the project schedule of each phase.

The project schedule is shown below:

ID	Task Name	Start	Finish	Duration
1	Planning Stage	6/21/2004	7/9/2004	3w
2	Literature Review	7/12/2004	8/13/2004	5w
3	Methodology/ Techniques Decision	8/16/2004	8/27/2004	2w
4	System Analysis and Design	8/30/2004	9/17/2004	3w
5	System Development and Implementation	11/22/2004	2/22/2005	13.4w
6	Documentation	6/21/2004	3/4/2005	37w

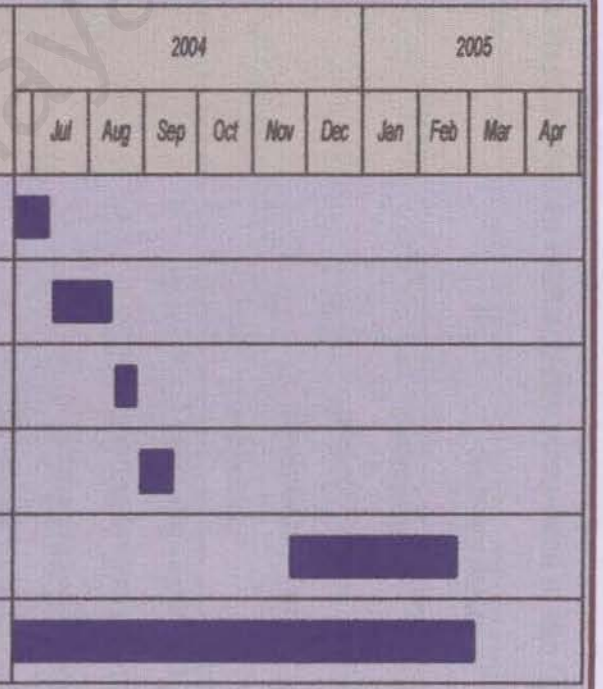


Figure 1.1 Gantt chart of Project Schedule

Table 1.1: Explanation on each activity mentioned in the Gantt chart

Activities	Sub-activities
Planning	• Determining problem definition, project aims, project scope, project limitations, and project development schedule.
Literature Review	• Thesis and paperwork review. • Existing system review. • Existing system demonstration. • Survey on techniques.
Methodology/ Technique Decisions	• Selecting the best and suitable technique. • Studying the chosen technique. • Determining the functional and non functional requirements. • Selection of development tools. • Hardware requirements.
System Analysis and Design	• Drawing system hierarchy, and flow chart. • Expected outcome.
System Development and Implementation	• Discover the modules in Molecule Visualization system. • Finding methods to segment and render molecular data. • Testing of the system with various molecular data.
Documentation	• Documenting each activities and findings in the research.

1.6 Report Layout

The dissertation of this project is organized in the following way:

Second Chapter: Literature review, which explains the research and analyze on the studies done about the research. This consists of possible techniques and reviews on the existing system besides elaborates on the project domain.

Third Chapter: Methodology part. This chapter is about the general steps taken in the overall molecular visualization research. Explains on functional and non-functional requirements, development tools which includes both software and hardware solutions.

Fourth Chapter: System analysis and design. This chapter reviews on the proposed solutions and results. In system analysis and design, it covers on system hierarchy, flow chart, and expected outcome. It also includes the combination processes of all parts in the system, which consists of its own executing functions.

Fifth Chapter: System Development and Implementation. This chapter reviews the different techniques in segmenting, rendering and various methods in molecule visualization. These are shown as step-by-step manual besides explaining the modules that are used in AmiraMol.

Sixth Chapter: Discussion and Conclusion. This chapter discussed on problems encountered and recommended solutions. It also reviews on the system strength and

limitations. Lastly, it discussed on future enhancements of this system before it came to a conclusion.

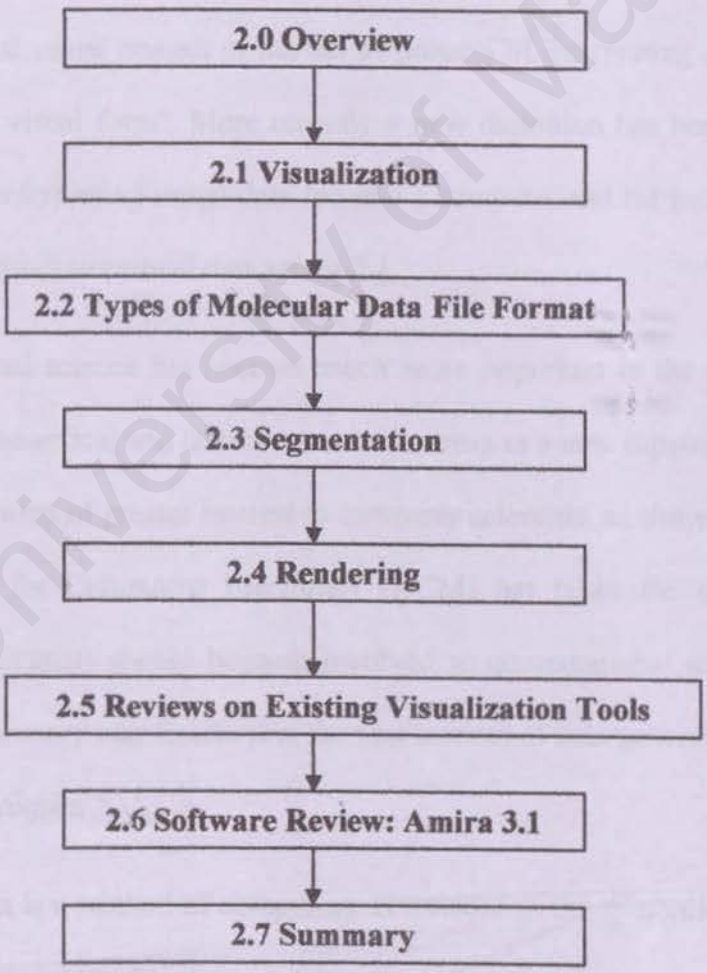
1.7 Summary

Overall, this chapter gives a broad overview of the project. It elaborates on the importance and feasibility of the project. Finally, it concluded with a project development schedule and report layout which is provided for further depiction of the project development.

CHAPTER 2

LITERATURE REVIEW

This chapter explains on the research and analyzes on the studies done about the research. This consists of possible techniques and reviews on the existing system besides elaborates on the project domain.



2.0 Overview

In chapter 2, it will discuss on visualization. Besides, it will explain briefly on the structures of DNA, various representations of molecules and types of molecular data file format available. Next, it will describe on segmentation and various techniques in molecule rendering. Reviews on existing system as well as software review are also included in this chapter.

2.1 Visualization

Visualization is an old term, which has received a large amount of interest in the computer science community. Visualization has previously been defined as the "formation of visual images or the act or process of interpreting in visual terms or of putting into visual form". More recently a new definition has been added: "A tool or method for interpreting image data fed into a computer and for generating images from complex multi-dimensional data sets" [2].

Computational science has become much more important in the sciences, joining the traditional theoretical and laboratory science areas as a new supporting methodology. It is also becoming of greater interest to computer scientists, as shown by the fact that the Association for Computing Machinery (ACM) has taken the official position that computer scientists should become involved in computational science. Visualization will be the primary way to interpret the vast amount of data generated by computational science techniques.

Visualization is a method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization

offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights [27].

2.1.1 Definition and History of Visualization

In the Oxford English Dictionary (5th edition), the word visualization is defined as *"to form a mental vision, image, or picture of (something not visible or present to sight, or of an abstraction); to make visible to the mind or imagination".*

Visualization has been used in maps, scientific drawings, and data plots for over a thousand years. Examples of this are the map of China (1137 ad.) and the famous map of Napoleon's invasion of Russia in 1812, by Jacque Minard. Most of the concepts learned in devising these images carry over in a straight forward manner to computer visualization and can be incorporated in courses in visualization.

Computer Graphics has from its beginning been used to study scientific problems. However, in its early days the lack of graphics power often limited its usefulness. The recent emphasis on visualization started in 1987 with the special issue of Computer Graphics, on Visualization in Scientific Computing. Since then there have been several conferences and workshops, co-sponsored by the IEEE and ACM SIGGRAPH, devoted to the general topic, and special areas in the field, for example volume visualization. There have also been numerous books and research articles on visualization in the past several years.

2.1.2 Purposes of Visualization

Visualization serves it purposes as listed below:

- Exploration/exploitation of data and information.

- Enhancing understanding of concepts and processes.
- Gaining new (unexpected, profound) insights.
- Making invisible visible.
- Effective presentation of significant features.
- Quality control of simulations, measurements.
- Increasing scientific productivity.
- Medium of communication/collaboration.

2.1.3 Disciplines in Visualization

Various disciplines in visualization are shown here:

Computer Graphics: The branch of science and technology concerned with methods and techniques for converting data to or from visual presentation using computers.

Computer Vision: The analysis of image content, the conversion of pictures into descriptions.

Image Processing: The analysis, manipulation, storage, and display of graphical images from sources such as photographs, drawings, and video. Image processing spans a sequence of three steps. The input step (image capture and digitizing) converts the differences in coloring and shading in the picture into binary values that a computer can process. The processing step can include image enhancement and data compression. The output step consists of the display or printing of the processed image. Image

processing is used in such applications as television and film, medicine, satellite weather mapping, machine vision, and computer–based pattern recognition.

Visual Perception: Visual perception is a form of processing involving a set of skills used to get visual information from the environment and integrate them with our other senses. This is done while incorporating all the integrated information with other things, such as past experiences, motivation and development, in order to derive understanding and meaning from what we are experiencing. It involves visual spatial skills, visual analysis skills and visual integration skills. This is also called by some visual information.

Art and Design: The products of human creativity consisting of aesthetics and style

[12].

2.1.4 Scientific Visualization

Overall, there are three types of visualization:

- Visualization in Scientific Computing (Scientific Visualization)
- Information Visualization
- Software Visualization

Scientific visualization is a new, exciting field of computational science spurred on in large measure by the rapid growth in computer technology, particular in graphics workstation hardware and computer graphics software. Visualization tools are beginning to impact our daily lives through usage in the arts, particularly film

animation, and they hold great promise for scientific research and education. When computer graphics is applied to scientific data for purposes of gaining insight, testing hypothesis, and general elucidation, it is called as scientific visualization.

Basically, scientific visualization means mapping from computer representations to perceptual or visual representations, choosing encoding techniques to maximize human understanding and communication.

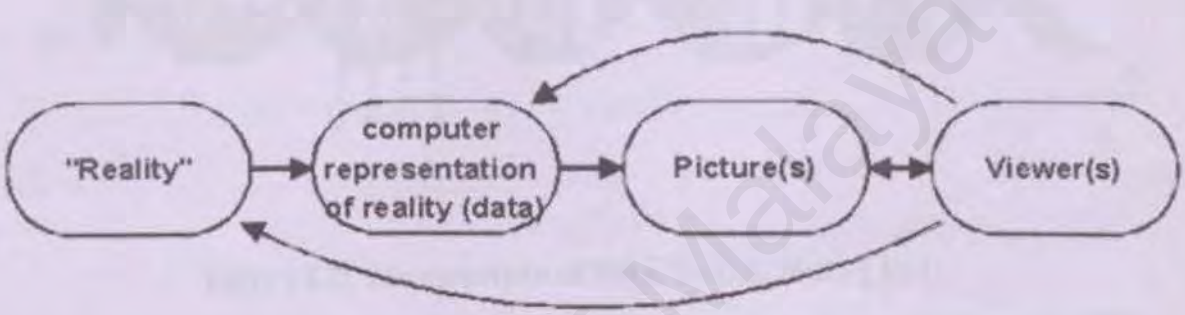


Figure 2.1: Overview of scientific visualization

2.1.5 Molecular Visualization

Before proceeding to the concept of molecular visualization, particularly on DNA molecules, first, one needs to have an understanding on the basic structure of DNA. DNA or deoxyribonucleic acid is double stranded sequences of four nucleotides that compose a strand of DNA, there are adenine (A), guanine (G), cytosine (C), and thymine (T). The chemical structure of DNA (the famous double- helix) was discovered by James Watson and Francis Crick in 1953. It consists of a particular bond of two linear sequences of bases. This bond follows a property of complementarity, adenine bonds with thymine (A-T) and vice versa (T-A), cytosine bonds with guanine (C-G) and vice versa (G-C) which is known as Watson-Crick complementarity. Each DNA

strand has two different ends that determine its polarity, the 3.'end, and the 5.'end. The double helix is an anti-parallel (two strands of opposite polarity) bonding of two complementary strands.

In recent years, many techniques have been developed in order to study and manipulate DNA in a lab, for various biological applications. [4]

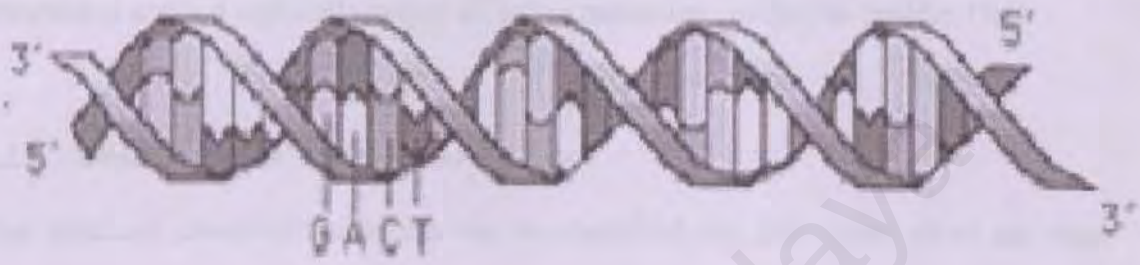


Figure 2.2: The structure of DNA Double Helix [13]

Molecules can be represented in many forms and has its hierarchy. Each level contains all members of the next level that are a part of it. The hierarchy is as follows:

- Chains – chain backbone information is stored.
- Secondary structure – a Helix, a Turn, or a Sheet.
- Residue – for example a nucleotide.
- Atom – the lowest tier.

Each level of the hierarchy is also assigned a geometric representation. An atom is represented by a single sphere with the atom's Van der Waal radius.

A residue is represented by a minimal single bounding sphere that encloses all of its component atoms.

For secondary structures, sheets and helices are represented by an appropriately sized geometric set of cylinders and helices. The orientation, length and radius of these are

determined by a least mean square fitting of the central axis to all of the atom centers in the helix or strands in sheet.

Chains are represented by the entire atom in the backbone of the chain.

Once the hierarchy is created, the user may interactively change the level of detail in the hierarchy by picking one of its tiers to visualize. Under static level of detail, this selection is applied uniformly across all entire molecular complexes besides DNA.

2.1.6 Representation of Molecular Models

The structural model of molecules can be classified into four types, there are wire-frame, ball stick, space filling, and stick model.

- The wire-frame model

It is the most simple and best known model, where the molecule is simply represented using colored vectors corresponding to the bonds.

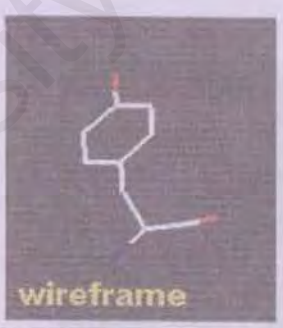


Figure2.3: Wire-frame model

- The ball & stick model

The molecule is represented with small spheres defining the atoms and with cylinders for the bonds (simple, double, triple).

Figure 2.4: Ball-and-stick model

- The space-filling model

Each atom of the molecule is represented by a colored sphere of Van der Waals radius. Each sphere is rendered with Gouraud shading.

Figure 2.5: Space filling model

- The stick model

In the stick model representation, each chemical bond is represented by one colored cylinder, but the atoms located at the endpoints are not explicitly drawn.

Figure 2.6: Stick model

2.1.7 Virtual Reality

The virtual reality (VR) is a way of enabling information users to participate directly in real-time to the three dimensional environments generated by computers. In a sense, VR solves the human-computer interface problem by removing the interface. The VR is the ultimate step from computer graphics towards reality, in combining several sensorial perceptions in the same time, vision, touch and the hearing. The VR thus allows exploring complex phenomena in a more natural and intuitive way and therefore to understand and analyze them. The chemistry is opening a broad new field for VR, as for example, in drug design. It is possible to study the way a substrate approaches the receptor of the enzyme and to determine the best path by filing the repulsion and attraction forces, calculated by a chemical model, through a force-feedback manipulator arm [3] [26].

Figure 2.7: Molecular Visualization in Virtual Reality Environment

2.2 Types of Molecular Data File Format

The following file formats are types of molecular data input:

- PDB
- MDL
- Tripos
- UniChem
- PSF/DCD (CHARMM, trajectories)
- amira's native file format ZibMolFile (trajectories)
- Z-matrix format

2.2.1 PDB (Protein Data Bank)

PDB is a file format for storing structures of biomolecules. The format is used for archiving molecules. Amira uses most, but not all, information supplied in a PDB file. Amira supports PDB files that follow the conventions of PDB format version 2.0 or higher. [20]

Syntax - Every level contains the field 'index', which is an internal identifier that does not necessarily correspond to the succession of the groups in the file. The 'comment' field contains the information of the field named 'atom name' in the PDB file.

The 'name' field of chains contains the chainID of the respective `ATOM` record. The chain name of atoms belonging to `atom` records whose `chainID` field is empty will be set to 'NIL'. The 'name' field of residues is composed of the chain name and the residue sequence number. If the residue is a het-group, the name will start with 'HET'. The field 'type' contains the residue names. If none is given, the field will be set to 'UNK' (for unknown).

Example of file format -

ATOM 441 CB VAL L 58 21.025 37.362 -2.515 1.00 8.15 1IGM 562

The atom will be known with index '441' and comment 'CB'. It will belong to the residue with name 'L58' and of type 'VAL' which is sited on the chain with name 'L'.

2.2.2 MDL (Molecular Design Limited)

MDL file formats used for saving chemical structures. The MDL file format has been used for several rendering projects. It is a low-tech format that defines syntax for

storing simple data in text and binary files which are arranged into nested 'chunks.' It also defines several standard types of data chunks.

Features - Among the MDL features are that it have a simple, practical way of storing models that was fast to read and write, very easy to use in programs, and reasonably space-efficient. The primary application of the format is storing models for realistic rendering, including complicated materials and large polygon meshes.

Syntax - There are two levels to the syntax of MDL files: The conceptual level and the file level. The conceptual level is what the application program sees through the I/O library, and is the same for text and binary files; the file level is what's actually stored in the file, and is different for text and binary files.

At the conceptual level, a file consists of four types of data, which are 4-byte float, 4-byte int, 8-character keyword, and variable-length string. A file is a sequence of chunks. A chunk consists of a keyword and a sequence of data items, each of which can be a float, an int, a string, or a chunk.

At the file level, a text file contains the identifying keyword followed by a sequence of chunks. A chunk is a keyword, followed by a sequence of data items, followed by the keyword `end`. Floats and ints are represented in scanf/printf format, with the restriction that a float has to have a decimal point. Keywords are represented as sequences of up to eight alphanumeric characters, and strings are delimited by double quotes.

Example file format (represented in the text format) -

```
mdlflA20

[ A diffuse blue sphere ]
sphr "racquetball"
   lmbrtn rgb 0.2 0.2 0.8 end end
   0.0 0.0 0.0 0.03  % center, radius
end
```

Here is the same file in the binary format (underscores are spaces).
2 words: `mdlf, lB20`
2 words: `sphr, ____`
1 word: 15
3 words: `racq, uetb, all\0`
2 words: `lmbr, tn__`
1 word: 6
2 words: `rgb_, ____`
1 word: 3
3 words: 0.2, 0.2, 0.8
4 words: 0.0, 0.0, 0.0, 0.03

Information stored – it stored the types of molecule and this information depends only on the keyword of the chunk the data is in. [21]

2.2.3 UniChem

The UniChem molecular modeling uses the UniChem file format (extension uni) software to save the structure of molecules and computational results. Amira uses all structural information supplied in the COORDINATES and BONDS blocks. Files that are exported by Amira follow file format version 4.0 and can be used with the UniChem software. The UniChem data format includes additional information not represented in PDB format.

Example of file format – Unichem molecule building structure is shown here:

```
JOB                          JOB
CALCULATION                  CALCULATION
CODE    MNDO94 2.0           CODE  DGAUSS 2.0
TYPE OPT                     TYPE OPT+prop
METHOD None                  METHOD LDF+BP
BASIS None                   BASIS DZVP
HAMILTONIAN AM1              HAMILTONIAN None
```

Information stored - Information found in this file includes the following:

- A version identifier.
- An indication of data type (`TYPE 1` is a molecular structure; `2` is a library structure).
- The number of atoms in the file (for example, 3).
- The name of the molecular structure.
- Bond information that is computed and used for graphic display input. [19]

2.2.4 PSF/DCD (CHARMM, trajectories)

This is a format used by CHARMM. It consists of two files, one containing the structural information (suffix .psf), and one containing the trajectory, i.e., the atom coordinates (suffix .dcd). Amira can read this format and translates the sections of the psf file into grouping levels. Binary PSF file format was introduced to save various information related to sets of DNA sequences.

Features - This file format is extendable, quite flexible and relatively straightforward. It allows for any number of data fields of any kind.

Syntax - The PSF file consists of nested structures. A header (16 bytes) which consists of size of the structure in bytes including header (4 bytes) and title (12 bytes). The index part consists of size of the index in number of pointers (2 bytes) and pointers to data fields, each with 4 bytes long. Data fields listed every pointer in the index points to the first byte of the data field. The address is just a relative offset from the beginning of

the structure. Fields may be either structures or simple fields of format containing the header (16 bytes) with data.

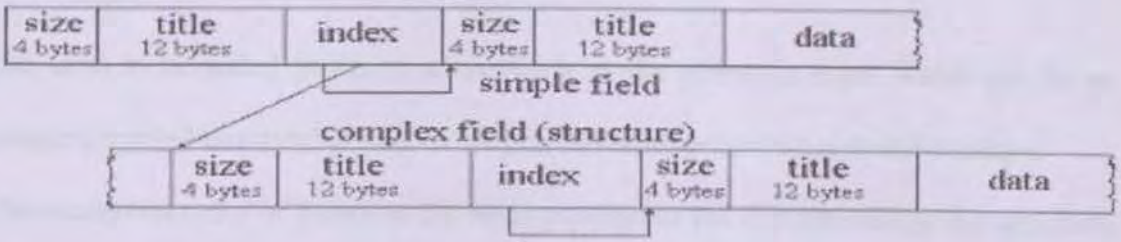


Figure 2.8: The syntax of PSF file format

Information stored –PSF save information about the entire data set, such as grouping of sequences and phylogenetic information and about every individual sequence, intron-exon structure, functional regions and chromatograms. [18]

2.2.5 Z-matrix format

Z-matrix file format is use to import molecules using internal coordinates.

Syntax – The syntax of this file format is shown below:

```
molecule_name
number-atoms
atom
atom  atom-r1  distance
atom  atom-r1  distance  atom-r2-angle
atom  atom-r1  distance  atom-r2-angle  atom-r3 dihedral
```

Information stored – Among the information stored in this file format are:

- Molecule-name

The name of the molecule. It will be used as the structure name in the UniChem application.

- Number-atoms

The total number of atoms in the file.

- Atom

The atom to be added to the structure, such as the reference atom. Atom can be an integer a symbol or a symbol and an integer. All atom occurrences must be unique

The sequential order of atoms in the atom position in the file determines the sequence number used in the UniChem file. This means that atom names in the internal UniChem structure will not exactly match the atom names in the external Z-matrix file when an out of sequence numbering is used for any atom name.

- Angle

The value of the angle formed by atoms. The angle must fall in the range $0 < angle < 180$ and is measured in degrees.

2.2.6 Tripos

The tripos file format is used to save Tripos Sybyl molecule information.

2.2.7 Amira's Native File Format ZibMolFile (trajectories)

The *zmf* file format is a structured file format developed for exchanging molecular data.

Basically all the molecule file format stores the following information:

i. Topology:

- Atom type
- Bonds/bond type
- Residue information, secondary structure, chain.

ii. Coordinate

- Atomic (in Cartesian)

2.3 Segmentation

The main purpose of segmentation is to find region that represents meaningful part within the object. For example, the molecular structure of the DNA contains nucleotides such as adenine, guanine, cytosine and thymine tied in some sort of bond. Through segmentation, only the specific nucleotide will be taken out and others will be ignored. It consists of two tightly coupled tasks, which are recognition and delineation. Recognition is the process of identifying roughly the whereabouts of a particular object in the image and delineation is the process of specifying the precise spatial extent and composition of the molecule.

2.4 Rendering

Rendering is the process of the creation of an image containing geometric models, using color and shading to give the image a realistic look. It creates an image on the screen from polygons, textures, lights, and other graphical information, as opposed to displaying pre-computed graphics and animation. In short, rendering means creating a 2D computer image out of 3D data by the graphics engine real-time. In rendering molecules, numerous modeling schemes have been used to represent and display molecules and their properties in computers. Some models are structural in nature include stick model, ball and stick model, wire frame model and the cartoon model. All these are different visual representation of an underlying hierarchy skeletal model of the positions of atoms, bonds, chains and residues in the molecule. A complementary approach to modeling molecule structure is to model properties of the molecule as three-dimensional field defined over structural model. This enables either extracting

isosurfaces of the fields or examining them using volume rendering [10]. Overall, there are many types of rendering according to molecule hierarchy. Among them are [30]:

i. Surface Rendering [7] [14]

Surface rendering is an indirect technique used for visualizing volume primitives by first converting them into an intermediate surface representation and then using conventional computer graphics techniques to render them. With surface rendering, only inner or outer surface of the molecules can be examined.

ii. Volume Rendering [15]

Volume rendering is a visualization technique used with three independent variables mapped to the three geometric dimensions. Each dependent data value is represented by a 3-D volume unit called a voxel, typically coded with color. Various sectioning techniques are used to reveal data values on the interior of the data space.

In rendering a volume using 3D textures, the first step is to load the volume data into a 3D texture. This is done once for a particular data volume. Then, it determines the number of slices before determining the desired viewpoint and view direction. Later, it computes a series of polygons that cut through the data perpendicular to the direction of view. Texture coordinate generation is used to texture the slice properly with respect to the 3D texture data. Texture transform matrix is used to set the desired orientation of the textured images on the slices. Upon completing this, render is carried out on each slice as a textured polygon, from back to front. A blend operation is performed at each slice; the type of blend depends on the desired effect. As the viewpoint and direction of

view changes, the data slice positions need to be recomputed and the texture transformation matrix need to be updated.

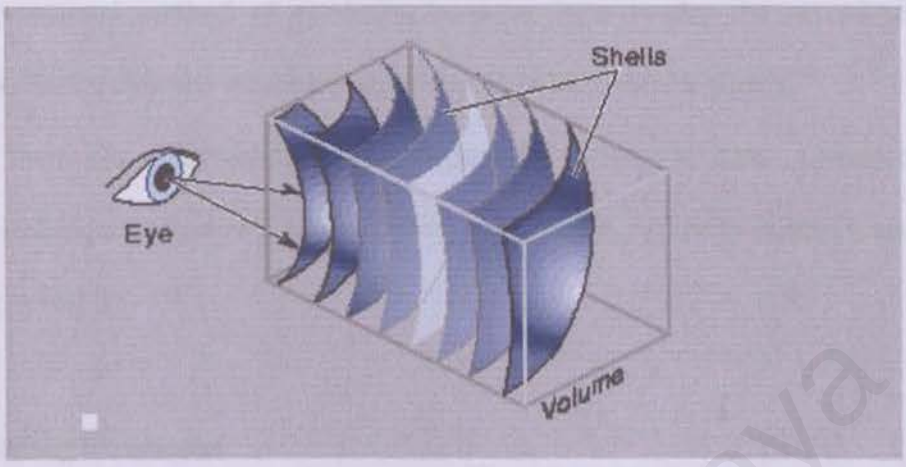


Figure 2.9: Slicing a 3-D texture

iii. Sphere Rendering

Used to render CPK (sphere) model. This method is used to render spheres in TexMol. It is the base used for the cylinder and helix rendering.

iv. Cylinder Rendering

Used to render ball and stick model. The cylinder uses a single quadrilateral as the underlying geometric primitive that uses three texture maps. A cylinder, however, is not rotation invariant, so its corresponding quadrilateral in clip space reflects the orientation the true cylinder had it been transformed to clip space.

v. Helix Rendering

Used to render secondary structures. Efficiently render per pixel lit helices and cylinders with minimal number of vertices. Helix rendering is composed of two quadrilateral primitives and are an extension of the cylinder rendering.

vi. Sheet Rendering

Used to render secondary structures. Sheets are known to consist of multiple strands. The conventional method of rendering a sheet is to render the set of strands as cylinders. Hence cylinder rendering algorithm is used to render sheets.

Overall, there are many rendering techniques for molecular data. Among them are shading techniques, shadows, transparency, ray tracing, radiosity method, and texture mapping. [16]

2.4.1 Shading techniques

Shading techniques consisted of Constant shading, Gouraud shading, and Phong shading.

i. Constant shading

Each facet of the object is illuminated by an average value for the entire polygon. This approach is fast and very simple, but it gives quite poor realistic results and non smooth surfaces. This is enhanced by the Mach effect, the intensity at the vicinities of the edges is overestimated for light values and underestimated for dark values.

ii. Gouraud shading

The Gouraud shading eliminates intensity discontinuities by interpolating the intensity for each polygon. It uses the normal vector at each vertex and edges of the polygon mesh which is obtained by averaging each normal of the facets sharing the same edge. From the picture below, on the left is a Connolly surface of the ferrocene in Gouraud shading, and on the right the same surface with flat shading.

Figure 2.10: Difference between Gouraud shading and flat shading

iii. Phong shading

The Phong shading is like the Gouraud shading based on an interpolation algorithm but this time, the interpolation is made by vectors. It uses the normals at each facet, the average normals at each vertex, and interpolates vectorially along the edges between to vertex, and then interpolates the same way between the edges along the scan-line. It involves very heavy calculation, as it has a normalization calculation at every step.

2.4.2 Shadows

Shadows techniques include self-shadows and cast-shadows.

i. Self-shadows

The self-shadows are the facets of objects visible from the point of view but invisible from the light source. For each facet, one has to determine if it is in the shadow (scalar product of the normal and source light is smaller than zero) or if it is lighted (scalar product greater or equal than zero).

ii. Cast-shadows

The cast shadows are determined by the projection on the scene of all lighted facets (visible from the light source). The projection is parallel if the light source is at infinity or in perspective otherwise. One calculates the shadow polygons projected over all the objects.

2.4.3 Transparency

Transparency consists of two types, there are non-refractive transparency and refractive transparency.

i. Non-refractive transparency

The non-refractive transparency, as indicated in its name, does not take into account the refraction of rays through the object. The interpolated transparency determines the value of the intensity of the ray at the intersection of two polygons. Given the following equation,

$$I_\lambda = (1 - k_{t1})I_{\lambda 1} + k_{t1}I_{\lambda 2},$$

in which kt1 transmission coefficient is the transparency measure of the polygon 1 and varies from 0 to 1. When the polygon is opaque the constant is equal to 0 and it is equal to 1 when the polygon is transparent. The value (1-kt1) corresponds to the opacity of the polygon, this term is similar to the value alpha of graphics workstations.

Below shows a series of the representation of the ferrocene with a blue Connolly surface, going from different stages of opacity (alpha), from the trivial case of alpha equals to 1 (completely opaque) to the trivial case alpha equals to 0 (completely transparent), with intermediate values of 0.75, 0.50, 0.25 and 0.10.

Figure 2.11: Series of the representation of the ferrocene with a blue Connolly surface, going from different stages of opacity

ii. Refractive transparency

The refractive transparency takes into account the refraction index of the crossed object. Its implementation is quite complicated as it necessities the use of ray tracing to determine the multiple reflections and refraction through the object. This technique is of very little interest in molecular graphics.

2.4.4 Ray tracing

Ray tracing is an illumination model that computes the color at a point, in terms of light directly emitted by light sources, and of light that reaches the point after reflection from and transmission through its own and other surfaces. This indirectly reflected and transmitted light is often called *global illumination model*. The recursive ray tracing is such a model. It gives therefore a very realistic vision of the rendered objects, as it takes into account shadows, specula reflections, refractive transparency, and non-punctual light sources.

This method gives very good results, but it is yet not widely used, in molecular graphics, as the need of real time manipulations is primordial. It is widely used in other fields of computer graphics for its great rendering and for its ability of being parallelized.

2.4.5 Radiosity method

The radiosity model is generally used in the representation of scenes including directional light sources. The radiosity is based on a global illumination model based on an energetic balance at the surfaces, for calculating diffuse reflections from object to object.

2.4.6 Texture mapping

Historically, the introduction of textures in the rendering techniques has been made by applying 2D textures on a 3D object to gain in realism, for example the classical case of displaying a picture of a house's facade, as a texture, on a single polygon to obtain a very realistic result with very few expense. It is quite obvious that such application of the texture is very limited in molecular graphics, although it can give quite interesting results.

The necessary components are:

- a texture 1D, 2D or 3D.
- a 3D object defined in vertex.
- a representation function putting together texture and vertex.

i. 1D texture mapping

One application of texture-mapping in molecular graphics is the use of one-dimensional textures for representing color-coded surfaces.

With the traditional Gouraud shading, implementation of the color-code is applied by assigning RGB color triplets to the vertices of the 3D geometry, and the pixel colors are generated by linear interpolation in RGB color space. This approach can lead to

artifacts and loss of information with sparsely tesselated objects and high contrast color-map.

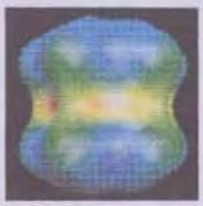


Figure 2.12: 3D geometry with color triplets

When using 1D texture mapping, the color ramp is stored in a 1D texture and the scalar property can be defined as the texture-coordinate for the surface vertices. The color interpolation is then made in the texture-space.

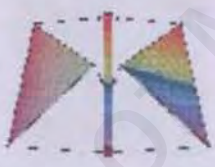


Figure 2.13: Color ramp stored in 1D texture

The resulting sharp transitions from one color to the other lead to sharp contour on the curved surface which is independent of the tessellation.

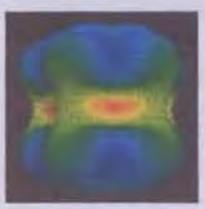


Figure 2.14: Sharp contour on the curved surface

This approach gives the opportunity of drawing contour lines by placing in the texture at certain contour thresholds distinct color.

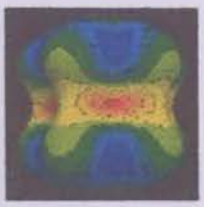


Figure 2.15: Drawing contour lines by placing in the texture at certain contour thresholds distinct color.

Isocontour lines can be drawn from this technique. The scaling between the lines can be adjusted in real-time in the texture ramp.

1D textures allow a much better rendering in the case of transparent surfaces.

Figure 2.16: Texture with Gouraud shading

Figure 2.17: Texture with 1D texture mapping

1D texture mapping allows clipping a surface in the texture space (scalar property) and not in the coordinates of the molecule.

Figure 2.18: Classical clipping plan in the molecule coordinates

ii. 2D texture mapping

A 2D texture can be applied through the traditional way by wrapping the texture over the surface, such as to put label on a ball-&-stick model of the ferrocene.

Figure 2.19: Labeling on a ball-and-stick model of the ferrocene

Another application of 2D texture mapping is high-quality surface-rendering. One can simulate Phong shading using a texture mapping procedure. It uses a high quality rendered sphere as a texture and by using the x- and y-components of the unit length surface normal as texture coordinates. Thus the interpolation in texture-space results effectively in an interpolation of the surface normals, so the result corresponds exactly

to a Phong-shaded picture with infinitely distant light-sources. As no shading before the texture-mapping step is necessary, this method is a very fast implementation of Phong-shading, even more now that texture handling is hardware implemented on high quality graphics workstations.

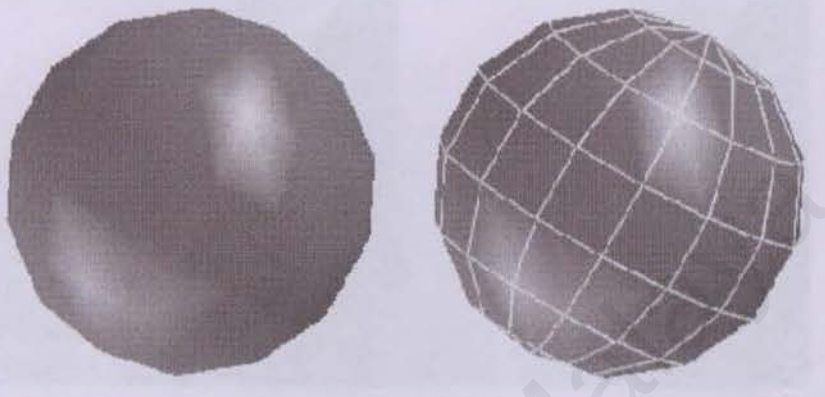


Figure 2.20: Gouraud shading with specular light artifact

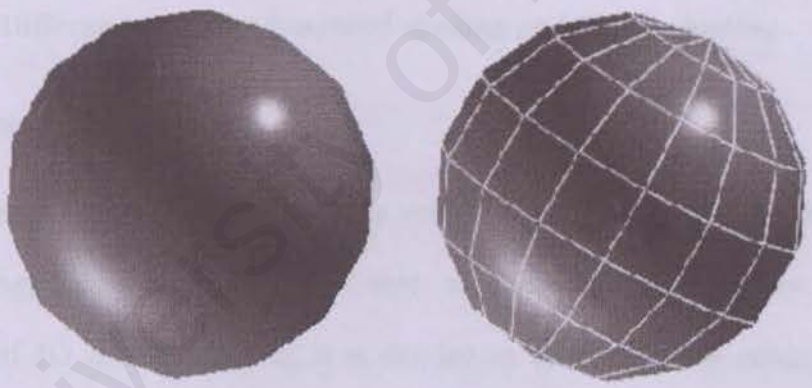


Figure 2.21: Phong shading through texture mapping

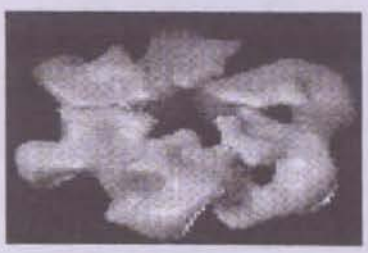


Figure 2.22: Biomolecular surface with Gouraud shading artifact

These close-ups show clearly the difference between Gouraud shading and texture-mapping implementation of the Phong shading.

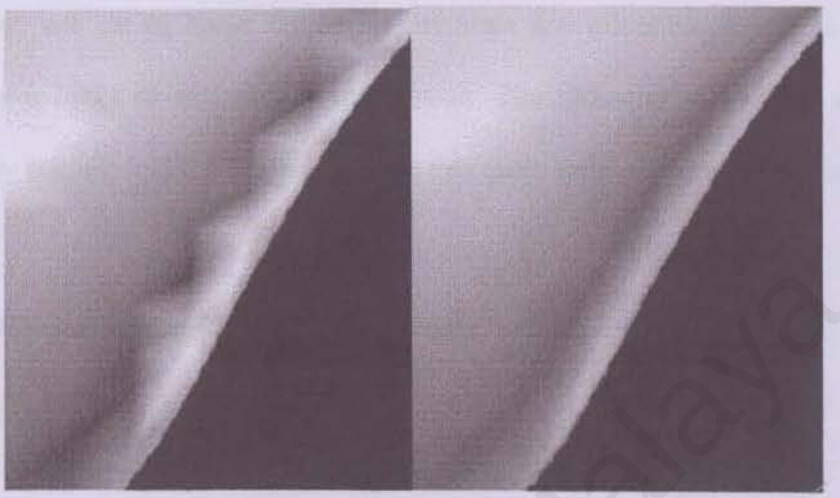


Figure 2.23(a): Gouraud **Figure2.23(b): Phong**

Figure 2.23 Difference between Gouraud shading and Phong shading

iii. 3D texture mapping

The relationship between 3D textures and volumes seems quite obvious, although the need for high-quality graphical hardware architecture is mandatory. A simple application of 3D texture mapping is to display an arbitrary plane cutting through a given volume.

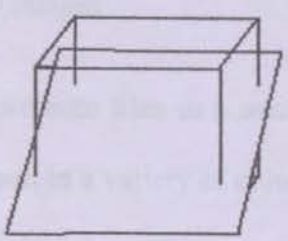


Figure 2.24: Arbitrary plane cutting through a given volume

This volume is directly related to a scalar property, for example the probability of finding a water molecule around a sugar molecule.

Short to say, not all of these rendering methods are often employed in molecular modeling as it lacks of real time manipulation. The radiosity method as well as the recursive ray tracing and the Phong shading are quite time consuming in the rendering procedure.

2.5 Review on Existing Visualization Tools.

2.5.1 RasMol

RasMol is a free program which displays molecular structure. It is available for Windows (RasWin), MacIntosh (RasMac), Unix, and VAX VMS and also (through ports by users) for NEXTSTEP and for Acorn Archimedes RISC OS. RasMol is a powerful educational tool for showing the structure of DNA, proteins and smaller molecules. It is also a powerful research tool. [22]

Features

- RasMol is a molecular graphics program intended for the visualization of proteins, nucleic acids and small molecules. The program is aimed at display, teaching and generation of publication quality images.
- RasMol reads in molecular co-ordinate files in a number of formats and interactively displays the molecule on the screen in a variety of colour schemes and representations.

- Currently supported input file formats include Protein Databank (PDB), Tripos' Alchemy and Sybyl Mol2 formats, Molecular Design Limited's (MDL) Mol file format, Minnesota Supercomputer Center's (MSC) XMol XYZ format and CHARMm format files. If connectivity information and/or secondary structure information is not contained in the file this is calculated automatically.

- The loaded molecule may be shown as wireframe, cylinder stick bonds, alpha-carbon trace, space filling (CPK) spheres, macromolecular ribbons (either smooth shaded solid ribbons or parallel strands), hydrogen bonding and dot surface. Atoms may also be labeled with arbitrary text strings.

- Different parts of the molecule may be displayed and colored independently of the rest of the molecule or shown in different representations simultaneously. The space filling spheres can even be shadowed. The displayed molecule may be rotated, translated, zoomed, z-clipped interactively using either the mouse, the scroll bars, the command line or an attached dials box.

Advantages

- RasMol can read a prepared list of commands from a `script' file (or via interprocess communication) to allow a given image or viewpoint to be restored quickly. RasMol can also create a script file containing the commands required to regenerate the current image.

- Finally the rendered image may be written out in a variety of formats including both raster and vector PostScript, GIF, PPM, BMP, PICT, Sun rasterfile or as a MolScript input script or Kinemage.

2.5.2 Chime [5] [8] [24]

An alternative free visualization tool is the Netscape (TM) 3.01 Plugin called Chime, supplied by MDL Information Systems, Inc. Chime is a free program to show molecular structure in three dimensions. Its images look like RasMol's because Chime is derived, in part, from RasMol. Chime differs from RasMol in that Chime sits directly on a web page (runs inside the browser as a plug-in), whereas RasMol is a stand-alone program (runs outside the browser, independently).

Features

- Chime has beautiful molecular representations.
- Consists of a menu interface which invites the user to step in and explore the molecular structure.
- Equipped with command and scripting language.

Advantages

- Its astonishing speed which allows movement of a spacefilled macromolecule on personal computers.
- Its ability automatically to save a script which when replayed will immediately regenerate any desired view of a molecule.
- Its ability to run in Windows, Macintosh, and Unix.
- Being free, with source code in the public domain.

2.5.3 LITHIUM [25]

LITHIUM is a contextual 3D molecular editor that allows any researcher to examine new candidate molecules within their biological context. LITHIUM's 3D molecular editor allows the creation and modification of molecular structures in the context of a protein facilitating the exploration of new molecules within the constraints of target biological systems.

Features

- 3D sketching for creation of new molecular structures.
- Editing of existing molecular structures independently or in the context of proteins.
- Full structure editing functions: atom/group addition, change atom/bond types, rotate bonds.
- Intuitive interface - 3D editor is designed to be as similar as possible to sketchers commonly used by chemists (such as ChemDraw™, MDL® ISIS/Draw).

Advantages

- LITHIUM supports all standard visualization formats for molecular structure including surface computations with property mapping. Visualization of biological macromolecules is enhanced by the addition of protein and DNA/RNA ribbons.
- Import and export a wide variety of chemical structure and related file formats (mol2, sdf, MDL mol, SMILES, etc.).

- Cut and paste structures from standard chemistry sketching packages (ChemDraw™, MDL® ISIS/Draw).
- State-of-the-art OpenGL graphics.
- Standard molecular rendering styles (lines, capped sticks, ball & stick, space fill).
- Compute and display molecular surfaces and protein ribbons.
- Property mapping onto surfaces and ribbons.
- Isosurface display from gridded data.
- Inter- and intra-molecular interaction display (bumps and hydrogen bonds).
- Comprehensive atom selection tools.

2.5.4 MOLCAD

MOLCAD is a sophisticated graphical rendering of molecular surfaces and properties.

Features

- MOLCAD™ exploits the power of the human eye by creating graphical images that reveal the properties of molecules essential for molecular recognition.
- Van der Waals and solvent-accessible surfaces can be calculated, and a broad range of properties can be mapped onto these surfaces: lipophilic potential, electrostatic potential, hydrogen bonding ability, local curvature, and distance.
- MOLCAD calculates and displays the surfaces of channels and cavities, as well as the separating surface between protein subunits, or between a receptor and ligand. For

proteins, MOLCAD creates ribbon displays that reveal the underlying secondary and tertiary structure, and maps onto these displays physical properties such as residue lipophilicity, flexibility based on atomic temperature factors,[5] and the packing density.

- Three-dimensional vector fields associated with molecular structures can be represented by MOLCAD. Cones drawn in space represent both direction and magnitude of an electrostatic field.

- Three-dimensional volume displays show all points of a three-dimensional property at once, with the density and color of the volume encoding the value of the property at any given point.

Advantages

- MOLCAD's rendering techniques allow the rapid calculation and display of property-coded surfaces for both small molecules and macromolecules.

- MOLCAD enable dynamic assessment of the interactions between molecules.

- MOLCAD allows real-time rotation and scaling of surfaces for molecules of any size, facilitating interactive examination of structures.

- Displays can be normalized to enable comparison of surface properties on different molecules.

- MOLCAD offers the most extensive set of surface displays available for understanding the physical properties of molecules and how they interact with other molecules.

- Multiple surface displays can be readily interpreted using interactive controls that restrict a surface area's display to regions within a specified property range.
- MOLCAD is unique in its ability to display the surface separating two molecules and the surface of intra-molecular channels. [17]

2.5.5 WebLab™ Viewer

Another visualization tool is WebLab Viewer, provided by Molecular Simulations, Inc. WebLab Viewer 2.0 is a program to display and manipulate molecular models. [23]

Features

- Sketch import from IsisDraw and ChemDraw with automatic 2D to 3D conversion.
- High resolution graphics for PowerMac as well as Windows/NT.
- Selectable / displayable amino acid residues.
- Hydrogen bond and bump monitor display.
- Color surfaces by property.
- Improved crystal and symmetry display styles.
- Side-by-side stereo and full screen views.
- Able to perform add or remove hydrogen.

Advantages

- Intuitive controls and the quality of the graphics, including clear labeling, and a wide variety of styles.
- The system displayed large proteins, DNA molecules, organic, inorganic and crystal structures with great clarity.
- Powerful tool for dividing a protein structure up into helices and sheets.

Disadvantage

- Some of the options appeared not to work, such as "Display" did not always bring up the menu, and it was difficult to select torsion angles.

2.5.6 DisMol Applet [6]

DisMol Applet is a Java based molecular viewer capable of displaying interactive molecular images over the Web.

Features

- The viewer is capable of reading XYZ and PDB type inputs.
- DisMol display input in either stick or CPK images.
- The program is available for free download.

Advantages

- Source code is provided.

From the reviews of all these molecular visualization tools, it can be concluded that each tool has its own strong features in providing analysis of molecular data.

2.6 Software Review : Amira 3.1

Amira is a three-dimensional (3D) visualization and modeling system originally developed at the Department for Scientific Visualization of Zuse Institute Berlin, Germany. It allows user to visualize scientific data sets from various application areas such as medicine, biology, chemistry, physics or engineering. Besides, it allows 3D objects to be represented as grids suitable for numerical simulations, notably as triangular surface and volumetric tetrahedral grids. Amira provides methods to generate grids from voxel data representing an image volume, and it includes a general-purpose interactive 3D viewer.

This three-dimensional visualization software product offers unparalleled techniques for creating complex data visualization. Among the techniques are direct volume rendering, iso-surfaces, innovative vector field visualization tools, image segmentation, surface reconstruction, surface simplification, and generation of tetrahedral grids.

Amira supports many types of import formats including raw formats, thus allowing immediate access to native data sets. [1] [11]

Data file format supported by Amira are as follow:

Table 2.1: File format supported by Amira

Geometry File Formats	Image File Formats	Molecular File Formats
• Altair Hypermesh • AmiraMesh • AmiraMesh LDD • AVS Field • AVS UCD • Binary raw data • DXF • FIDAP NEUTRAL • Fluent/UNS • HxSurface • IDEAS Universal • OpenInventor iv • Plot 3D single structured • STL • VRML Besides, it also support for other file formats upon request such as Radioss from Mecalog, Madymo from TNO Automotive.	• ACR-NEMA • Analyze 7.5 • Analyze AVW • BMP • Binary raw data • Bio-Rad Confocal • Encapsulated PostScript EPS • DICOM • Icol Colormap • JPEG • Leica 3D TIFF • Leica experiments.lei • Leica Slice Series info • Metamorph STK • PNG • PPM/PGM/PNM • PSI • PLY • SGI RGB • Stacked Slices • TIFF • Zeiss.Ism	• MDL • Protein Database (PDB) • PSF/DCD CHARMM • Tripos • Unichem • ZIB Molecular File Format ZMF

Amira system can be categorized into three major components, there are:

- AmiraDev™
- AmiraMol™
- AmiraVR™

2.6.1 AmiraDev™

This edition of Amira allows user to extend the base Amira application by adding custom data types, visualization or processing modules, and input/output routines through a well-documented C++ API. AmiraDev can be the ideal development and research platform for 3D data processing and visualization tasks as it is object-orientated based and support powerful standard software layers such as OpenGL™, OpenInventor™, and Qt™.

2.6.2 AmiraMol™

AmiraMol™ module consists of powerful tools for molecular visualization compared to the existing Amira platform. It contains support for standard molecular file formats covering static and dynamic molecular data, tools for visualization processing and analysis of static molecules and molecular trajectories. AmiraMol™ module of Amira combines base system capabilities for 3D data visualization, such as hardware-accelerated volume rendering, with specific tools for molecular visualization and data analysis. Examples of AmiraMol™ functionality are molecular surfaces, sequence alignment and configuration density computation. AmiraMol™ is suitable for researches in chemistry, bio-chemistry and pharmaceutical industry.

2.6.3 AmiraVR™

AmiraVR™ is designed for virtual reality research. AmiraVR™ features include head-tracking, support for tracked input devices, as well as multiple video output channels and multiple graphics pipes. Besides, there are also multi-wall supports such as CAVE® or Holobench® systems and integration of user interface components into the virtual 3D scene. On part of that AmiraVR™ also support active and passive stereoscopic displays and software soft edge blending.

2.6.4 Features of Amira 3.1

The capabilities of Amira underlies on its features. Among the strong features of Amira are as follow [28]:

i. Data Import – Data can be loaded directly into Amira. A large number of standard file formats are supported. Among the data types are:

- Polygon models and finite element data.
- Scalar and vector fields.
- Scattered data formats.
- 2D image data and stacked image data.

ii. Slicing and Clipping – In slicing and clipping, Amira lets user to explore 3D imagery looking at single or multiple orthographic or oblique sections. Clipping data enable user to uncover hidden regions. Among the features are:

- Semi-transparent slice display.
- Psuedo-coloring for overlaying sections with functional data.
- Supports display of iso-intensity lines.

- Interactive clipping planes.

iii. Data Manipulation and Filtering – Amira is equipped with a variety of digital filters, editors and data processing modules which supports efficient 3D image manipulation.

- Filters: median, unsharp masking, histogram equalization, Laplace, Gauss, Lanczos, Sobel, etc.
- Interactive crop editor.
- 3D image stitching.
- Resampling and resolution control.
- Arbitrary arithmetic operations.
- Arbitrary contrast and color mapping.

iv. Surface Rendering – In surface rendering, Amira enable user to display and explore detailed 3D surface models. A multitude of drawing styles and color schemes have enhance a more meaningful and informative visualization.

- Support rapid iso-surface generation.
- Robust and detail-preserving reconstruction of surface models from segmented image data.
- Support physically-based rendering of semi-transparent models.
- Consists of wire-frame display.
- Pseudo coloring on the surface.
- Display of partial surfaces.
- Powerful multi-style rendering.

v. Volume Rendering – Amira supports volume rendering task by allowing a direct visualization of 3D image data using a physically based light emission/absorption model.

- Consists of interactive manipulation of transfer function.
- Intuitive volume editing by drawing in the viewer.
- Resolution control and region of interest.

vi. Image Segmentation – Another features in Amira is it allows image segmentation. Segmentation assigns labels to individual pixels in the image data in order to identify and distinguish different materials or structures. Segmentation is the prerequisite for accurate 3D model generation as well as for advanced data analysis tasks.

- Amira supports automatic threshold segmentation.
- Consists of powerful segmentation editor for interactive labeling of structures.
- Equipped with brush tool, lasso tool, magic wand (for region growing), and contour fitting tool.
- Rapid processing of large datasets by interpolation between key slices.
- 3D wrapping tool for organic shape interpolation from very few orthogonal slices.
- Specific segmentation filters, including island removal, smoothing, interpolation.

vii. Advanced Polygonal Models – Polygon post-processing tools generate compact surface models of the highest possible quality, which is important for efficient processing and accurate data analysis. Using Amira, user can perform the following:

- Generate smooth and consistent surfaces.
- Model simplification.
- Flexible grouping of sub-structures and material types.
- Surface smoothing.
- Surface editing.
- Mesh quality analysis.

viii. Large Data – Large datasets can be processed within Amira at interactive speed.

- Consists of fast interactive OpenGL rendering exploiting latest graphics hardware.
- Enable easy access to region of interest (ROI) in a very large datasets.

ix. Flow Data – Advanced vector field visualization allows user to display results of flow simulation or measurement within the 3D model.

- Interactive vector arrow display.
- Illuminated stream lines.
- Stream ribbons.
- Colorized flow textures.
- 3D stream surfaces.
- Provides integration with all other types of data and visualization.

x. Data Analysis – Amira allows probing, measuring, counting and other statistical modules to quantify densities, distances, areas, volumes, etc. Data analysis can be performed by following:

- Histogram analysis.
- Probing of data values in a point/sphere, or along a line/curve.

- Measurement of angles, distances, areas, volumes.
- Counting of connected components.
- Statistics output for each material and per connected component.
- Curvature analysis on a surface.
- Correlation analysis tools.

xi. Viewing and Navigation – This system can display single or multiple datasets in a single or multiple viewer windows, and navigate freely around or through these objects.

- It consists of multiple independent or synchronized viewer windows.
- Arbitrary view points inside or outside the object.
- It supports stereoscopic viewing.
- Restrict data display to user-defined region of interest (ROI).
- Support arbitrary combination of different visualizations in the same 3D view.

xii. Scripting – Scripting provides a flexible way of customizing Amira and automate tasks without the need for C++ programming. Among the features are:

- Full control over all modules and viewers through Tcl script interface.
- Easy definition of script objects with graphical user interface.
- Streamlining and customization of routine tasks.

xiii. Presentation – In presentation, Amira supports the following:

- High-resolution tiled or off-screen rendering for print quality images.
- Alpha channel output for seamless integration of images in other presentation material.
- Direct movie file export.

- User-defined animation of visualization parameters.

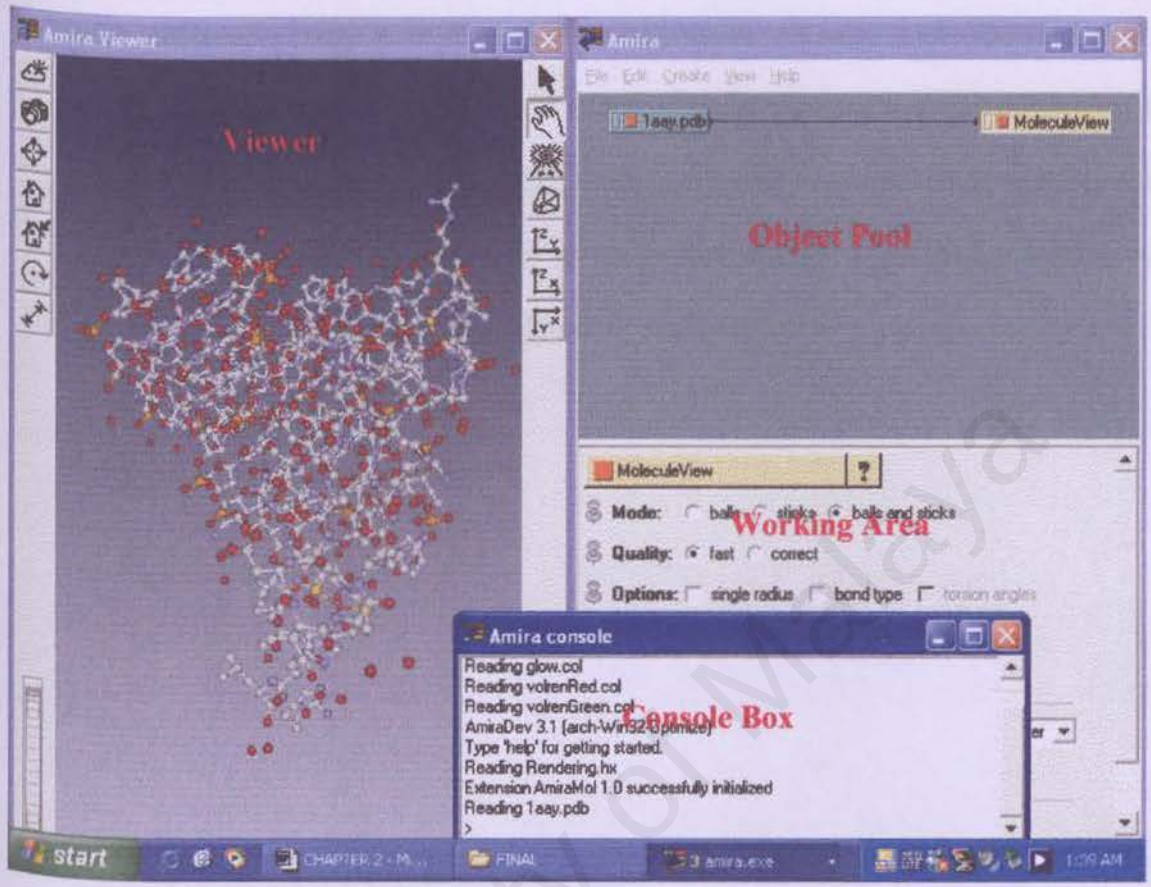


Figure 2.25: Amira 3.1 User Interface

2.7 Summary

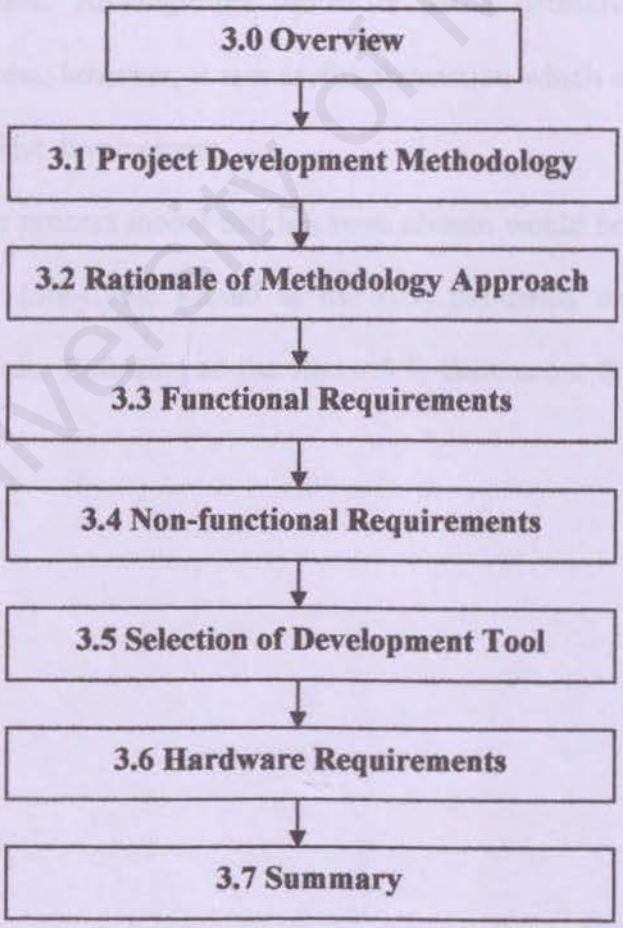
In conclusion, molecules appeared in hierarchy form in which the residue and atom level can be represented in various structural models, such as stick-and-balls, wireframe, etc. Meanwhile, in molecular data file format, though some format stores information that is not listed in other format, however, mostly they includes core information of the molecules such as its topology and coordinate. Meanwhile, there are various types of rendering techniques in which problem occurs in some techniques

would be of performance as some are quite time consuming if large molecule data are to be rendered. Thus, it is important to take into account prior knowledge while determining methods that are specialized to particular types of molecule that yield best performance. Regarding existing tools, there is no clear point on which is the best tool in providing analysis on molecules as each tool has its own strong features with different approaches in providing a dynamic visualization on molecular data.

CHAPTER 3

METHODOLOGY

This chapter is about the general steps taken in the overall molecular visualization research. Explains on functional and non-functional requirements, development tools which includes both software and hardware solutions.



3.0 Overview

A system development methodology is a very formal and precise system development process that defines a set of activities, methods, best practices, deliverables and automated tools for system developers.

Methodologies ensure that a consistent, reproducible approach is applied to all projects. It reduces the risk associated with shortcuts and mistakes. Finally, methodologies produce complex and consistent documentation from one project to the next.

3.1 Project Development Methodology

A system development model gives a standardized and systematic approach to the project development. Although the model is not a definitive description of the development process, however, it is a useful abstraction which can be used to explain approaches to project development.

In this project, the process model that has been chosen would be Waterfall Life-Cycle Model. Waterfall Life-Cycle Model is the first published model of the software development process. Referring to the figure 3.1, there are a few feedback loops for each phase.

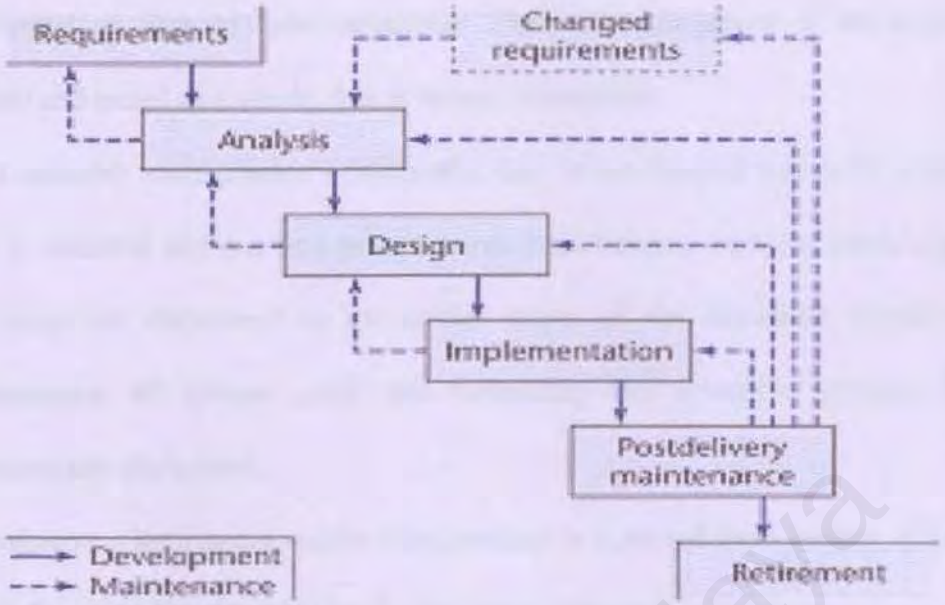


Figure 3.1 : Waterfall Life-cycle Model.

An overview of the phases is shown below:

i. Requirement phase – During the requirement phase, the concept is explored and refined, and user's requirements are elicited.

ii. Analysis (Specification) phase – The user's requirements are analyzed and presented in the form of the specification document. This phase is sometimes called the specification phase. Normally, at the end of this phase, a plan is drawn up, describing the proposed software development in full detail.

iii. Design phase – The system design phase partitions the requirements to either hardware or software. It establishes overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships.

iv. Implementation phase – During this stage, the software design is realized as a set of program or program units. These various components undergo coding and unit testing

is performed on each of them separately. Then, the component of the product are combined and tested as a whole, this is termed integration.

v. Post delivery maintenance – Normally this is the longest life-cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in the earlier stages of the life-cycle, improving the implementation of system units and enhancing the system's services as new requirement are discovered.

vi. Retirement – Retirement occurs when product is removed from service. This occurs when the functionality provided by the product no longer is of any use.

The characteristic of Waterfall model are:

- No phase is complete until the documentation for that phase has been completed. This carries over into modifications, if the products of an earlier phase have to be changed as a consequence of following a feedback loop, that earlier phase is deemed to be complete only when the documentation for the phase has been modified and the modifications have been checked for quality.
- Inherent in every phase of the Waterfall model is testing. Testing is not a separate phase to be performed only after the product has been constructed, nor it is to be performed only at the end of each phase.
- Characterized by feedback loop, thus, in maintenance, it is necessary to ensure that the modified version of the product still does what the previous version did and also have to do it correctly. In fact, it must also satisfy any new requirements added [9].

In implementing the Waterfall methodology for this system development, these following steps are taken:

- The system requirements are defined, which are gathered through various discussions with the project supervisor and also by reviewing some existing systems that are currently available.
- Next, I will analyze the needs for this system. Within this phase, special tools and techniques can help to make requirements determination. I used system hierarchy and flow chart in exploring the system in more detail.
- Designing the recommended system would be the next step. All the information gathered earlier would be used to accomplish the research of this system.
- System are thoroughly tested and evaluated, noting its flaws. Remarks on system performance will be collected and documented.
- This process is repeated in the same manner throughout the entire development process.
- Preceding steps are iterations, which are finally concluded when the system has fully functioned and satisfied all the requirements. Final product is to be delivered.

3.2 Rationale of Methodology Approach

The Waterfall model has been chosen to aid in the development of this project because of its many advantages. There are several reasons on why I prefer this model compared to other system models.

- It has an iterative approach in modeling processes whereupon each phase is revised continuously and flaws detected during revision period can be corrected along with the development progress.
- As we all know, requirements always change due to its need. Through Waterfall model, it allowed the requirements changes to be done easily. Each phase in Waterfall model is never accomplished as a separate step. In fact, several activities can occur simultaneously and activities may be repeated. This would give priority to users' need.
- The Waterfall model is documentation driven, in which limitations occurred in each phase and method in solving them are documented for future reference. Whenever a problem occurs in system development, I can always refer back to the documentation to find its fault.

3.3 Functional Requirements

Functional requirements are the actions or features, which expected by the users and stated by them to be incorporated into the system. The system is considered incomplete if any necessary function is not included.

Typically, functional requirement includes statements of services the system should provides, how the system should reach to particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

The functional requirements for this project are:

- **Display molecule** – The system must be able to load in various data file format. It should be able to displays the molecule in a variety color schemes and three-dimensional representations. The loaded molecule may be shown as wire frame, cylinder stick bonds, and space filling (CPK) spheres. The displayed molecule may be rotated, translated, and zoomed using mouse or scroll bar.
- **Editing** – This system allows user to change the color of the atoms for display purpose.
- **Selection, Labeling and Color Editing** – Different parts of molecule may be displayed independently to distinguish them with the rest. This can be done with selecting atom, residue, secondary structure or chains. Label can be assigned on each segmented part to differentiate each of them.
- **Alignment of Molecule** – Alignment is a classical task in molecular science. When two molecules are to be compared based on Cartesian atom coordinates, alignment is necessary to eliminate differences. Alignment need to be carried out before the computation of mean molecule and configuration density for visualization purposes.
- **Molecular surfaces** – Computation on molecular surfaces, and partial surfaces would be done through this module.
- **Generate output** – Each visualization tasks must be able to generate an output or to display them, either in graphic form in the "viewer pane" or in text/string/numerical figures in "working area" of AmiraMol 3.1.

3.4 Non-Functional Requirements

Non-functional requirements are essential definitions of system properties and constraints under which a system must operate. The constraints include timing constraints, constraint on the development process, standards and so forth. Among the non-functional requirements in this project are:

- **Reliability** – To achieve the objective of providing this project as an extra tool in aiding researchers, it has to be of high reliability, and failure rate is to be minimized.
- **Consistency** – The system must be able to produce a precise and consistent output each time it is run. This will increase the reliability factor as well.
- **Response time** – There are two arguments regarding this issue. If the system is used as a learning tool for students, time might not be much of a deal. However, if it is to use as a tool in researching, the system must be able to deliver the correct visualization in time, thus reducing cost by minimizing delay.
- **Correctness** – This method used should be able to perform tasks that satisfy users' mission objectives.
- **User friendly** – The navigation of data icons and modules icons (in Object Pool in AmiraMol) while performing the different visualizing methods should be as simplified as possible to ease researchers in performing their own tasks.

3.5 Selection of Development Tool

This research will be implemented in the 3D visualization and 3D image analysis system AmiraMol 3.1.

The supported platforms that enable this software to operate are as follow:

- Microsoft Windows 98/ ME/ NT4/ 2000 XP
- SGI Irix 6.5.x
- HP-UX 11.0
- Sun Solaris 8
- Linux Redhat 8.0

However, in this project it will be running on Windows XP Professional Edition.

3.6 Hardware Requirement

The hardware requirements are as follow:

- Intel Pentium 4 Processor.
- 128 MB RAM.
- 16 MB graphics card.
- SVGA monitor with 1280x1024 screen resolution.
- Mouse and keyboard.

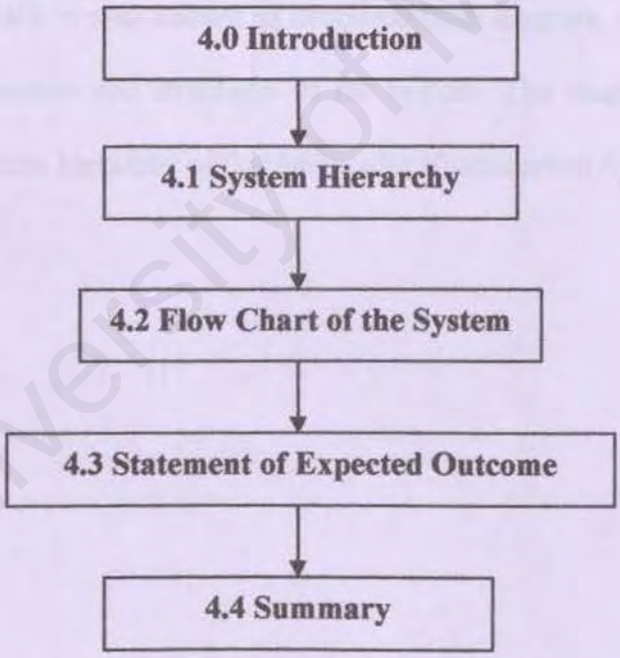
3.7 Summary

Overall, this chapter explained the methodology used in this project development which is the waterfall life-cycle model. It also describes the rationale of this approach. Apart from that, functional and non-functional requirements of the system are also included. Next, it listed out the software requirements such as the development tools, programming languages and operating environment that are to be used. Lastly, the hardware requirements such as machines and devices that are necessary for the system to run are stated.

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

This chapter reviews on the proposed solutions and results. In system analysis and design, it covers on system hierarchy, flow chart, user interface design, and expected outcome. It also includes the combination processes of all parts in the system, which consists of its own executing functions.



4.0 Introduction

System analysis is a problem-solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose.

System design is a complementary problem solving technique that reassembles a system's component pieces back into a complete system. This may involve adding, deleting, and changing pieces relative to the original system. [29]

4.1 System Hierarchy

Hierarchy chart, which is also known as decomposition diagram, shows the top down functional decomposition and structures of the system. The diagram shown on next page depicts the system hierarchy of this Molecular Visualization System.

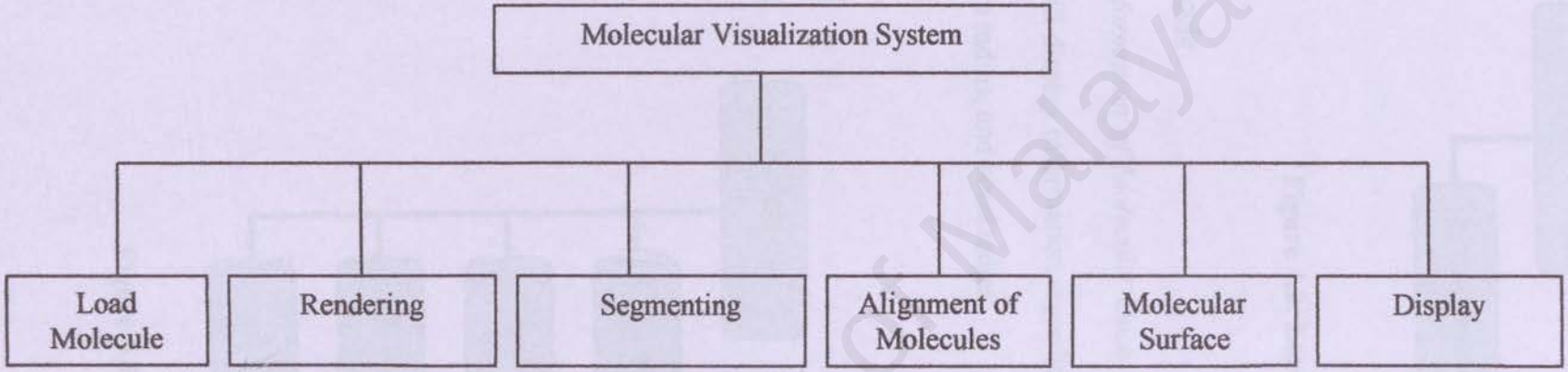


Figure 4.1: System Hierarchy of Molecular Visualization System

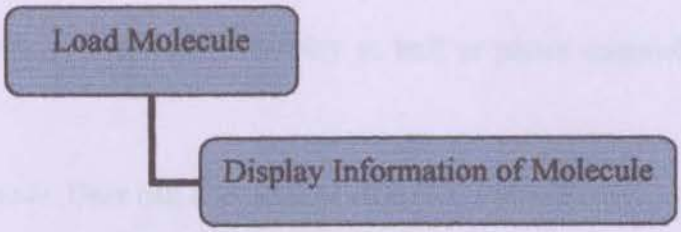


Figure 4.2: Load Molecule module

Load Molecule

Display Information of Molecule: Once the molecule is loaded into the system, this function will display information regarding that particular molecule such as number of atoms, atom radius, and bond radius.

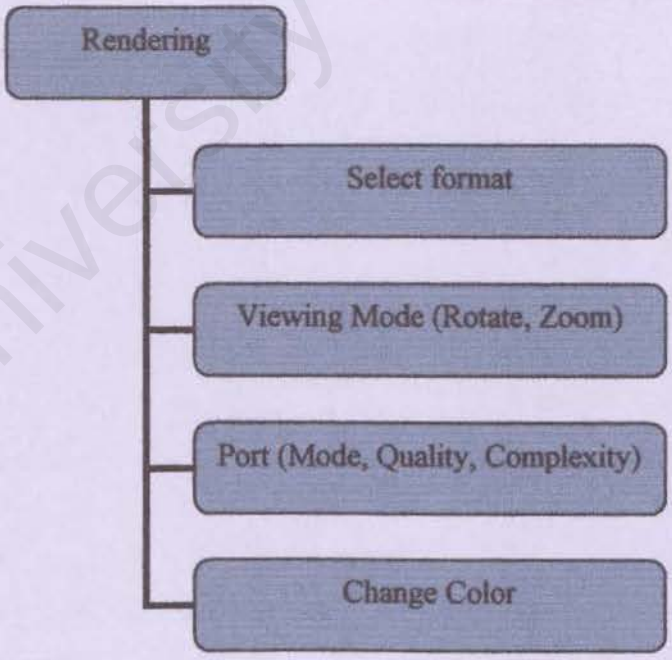


Figure 4.3: Display Molecule module

Rendering

Select format: Atom can be display as ball or plates meanwhile bonds as cylinder or line.

Viewing mode: User can choose type of mode, either zoom to get a closer view or rotate the molecule in different angle.

Port: There are basically three types of port, which are mode port (consist of ball, sticks, ball and sticks), quality port (either 'fast' which cater for rendering large molecules or 'medium' for a slower rendering), and complexity port. For complexity port, in order to allow interactive rendering, the default complexity of the scene is sufficient, but to make screen shots in viewing small molecules, user might need to increase the complexity. The complexity port is displayed only if correct quality is selected.

Change color: The default color scheme is according the atom type, however it enable user to change the color as well.

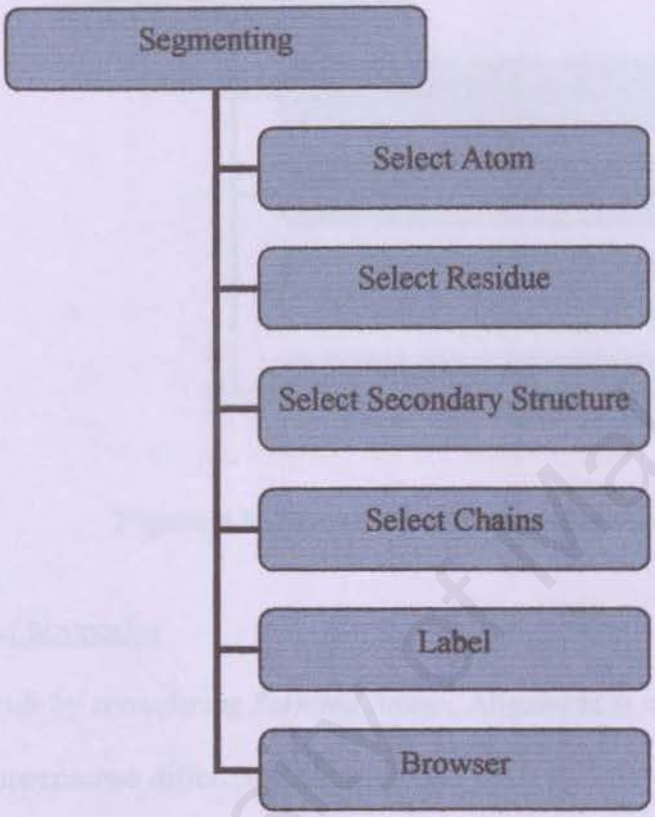


Figure 4.4: Selection and Labeling module

Segmenting

Selection: User can select atoms, residue, secondary structured and chains as desired for further analysis.

Label: It is necessary to label the molecule in the viewer to easily keep track of certain groups that are interested in.

Browser: Enable user to combine different viewing modules into one viewer.

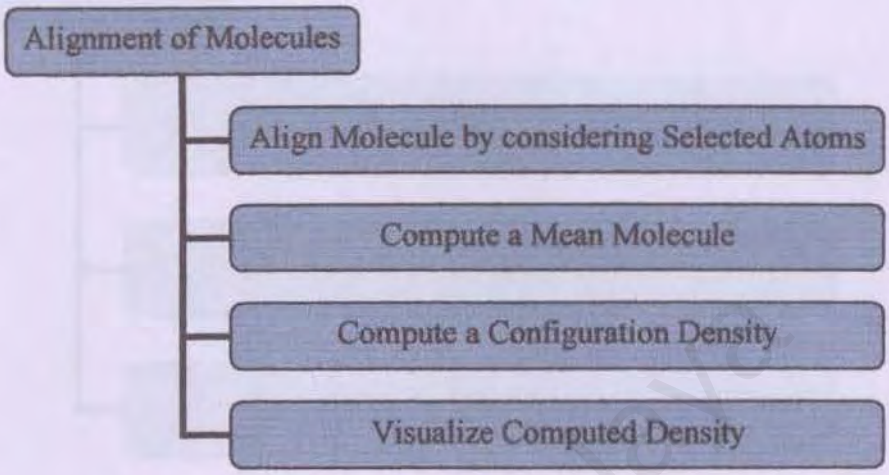


Figure 4.5: Alignment of Molecules module

Alignment of Molecules

Align Molecule by considering Selected Atoms: Alignment is used to establish degree of similarity between two different molecules.

Compute a Mean Molecule: A function to determine the orientation, length and radius by computing a least mean square fitting of the central axis to all of the atom centers in the helix.

Compute a Configuration Density: This function computes configuration density for visualization purpose.

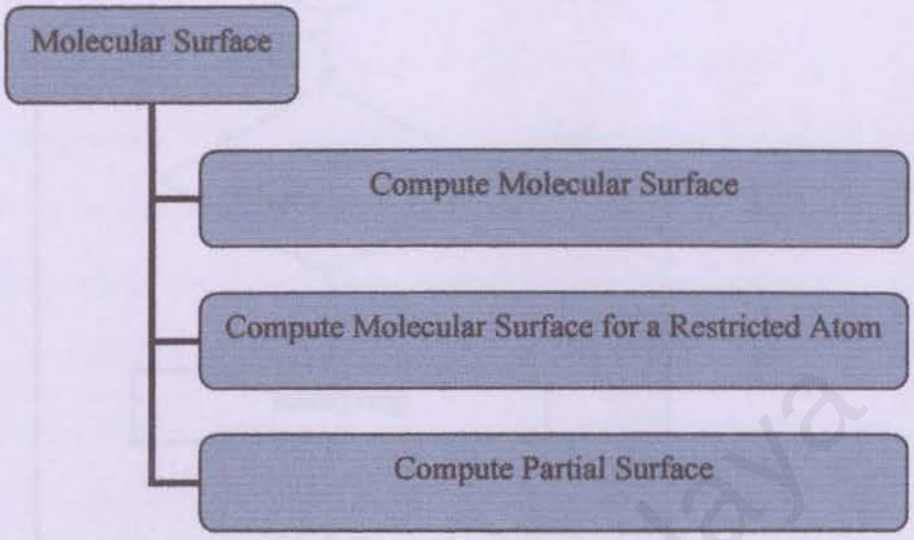


Figure 4.6: Molecular Surface module

Molecular Surface

Compute Molecular Surface, Compute Molecular Surface for a Restricted Atom and Compute Partial Surface: This allows user to perform computation on surface contribution for visualization purposes.

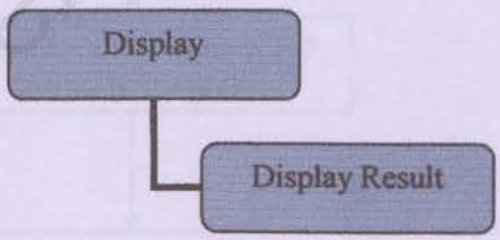


Figure 4.7: Display module

Display

Display Result: A function to view result from various computation, measurement and analysis.

4.2 Flow Chart of the System

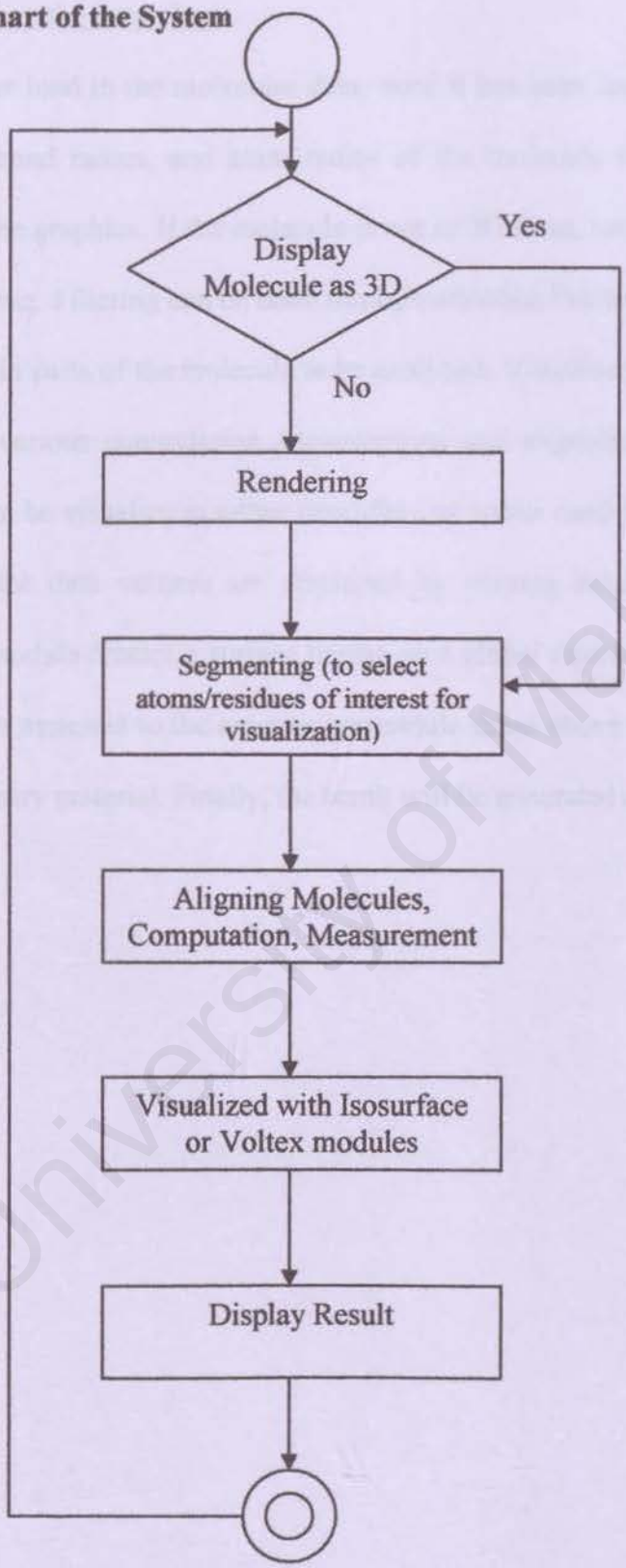


Figure 4.8: Flow chart of Molecular Visualization System

Explanation on the flow chart

First the user load in the molecular data, once it has been loaded, information such as file name, bond radius, and atom radius of the molecule will be displayed besides displaying the graphics. If the molecule is not in 3D form, rendering will be carried out before filtering. Filtering can be done through selection function and labeling. This is to enable certain parts of the molecule to be analyzed. Visualization is the next step where it involves various computation, measurement and alignment. After these steps, the molecule can be visualize in either isosurface or voltex modules. In voltex module, the Voxels of the data volume are displayed by varying brightness and transparency. Isosurface module creates a surface basing on a global threshold. All Voxel below this threshold are assigned to the exterior meanwhile those above the threshold are counted to a preliminary material. Finally, the result will be generated and displayed.

4.3 Statement of Expected Outcome

The molecular visualization research is going to be carried out in four-month duration, from November 2004 until February 2005. It is projected that various methods in molecular visualization will be discovered hence enable to segment, render and visualize static and dynamic molecular data particularly on DNA, protein and alkane molecules. It is hoped that the outcome is a very dynamic system which is able to deliver a virtual interaction between users and the system itself, thus aiding researchers in most of their molecular analysis.

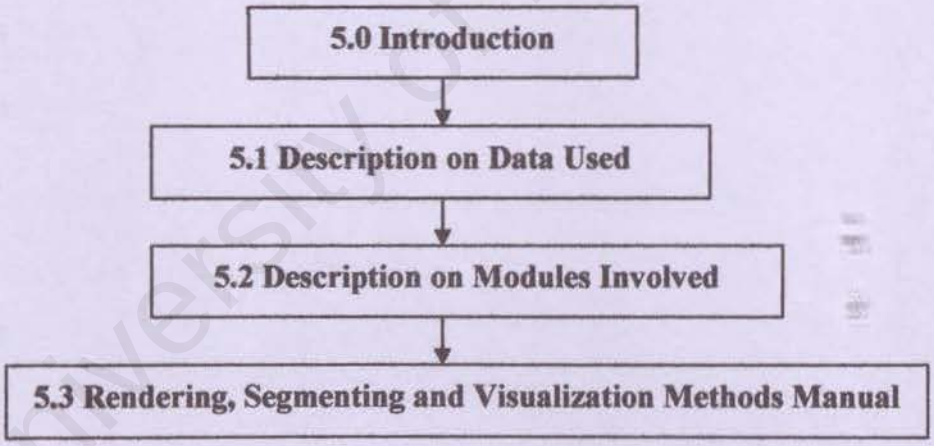
4.4 Summary

Overall, this chapter shows the system analysis and design of molecular visualization system. Since this system doesn't have a database, this chapter only elaborates on system hierarchy, and the system flow chart. Besides, a statement of expected outcome is included as well. This would probably give an overview idea on what the actual system would be.

CHAPTER 5

SYSTEM IMPLEMENTATION

This chapter reviews the different techniques in segmenting, rendering and various methods in molecule visualization. These are shown as step-by-step manual besides explaining the modules that are used in AmiraMol.



5.0 Introduction

This chapter explained the implementation part of Molecule Visualization. First, it will explain briefly on the molecular data that are used in performing the molecular visualization tasks. The second subchapter will describe on modules of AmiraMol 3.1 that are related in carrying out rendering and segmenting as well as visualization. These modules will be sorted alphabetically. Lastly, in the third subchapter, a manual regarding the methods performed in this research will be included. This will be guided with screenshots of each activities and processes.

5.1 Description on Data Used

Overall, the classification of molecules is divided into 4 categories, they are:

i. Simple Molecules
   - Gases
   - Foods
   - Fuels
   - Fats
   - Soaps

ii. Polymers
   - Synthetic (such as plastic, rubbers, acrylates, polyester, nylon)
   - Natural Polymer (such as hair wool, silk, sugar, starch, cellulose, DNA)

iii. Senses
   - Taste
   - Smell
   - Vision (such as receptor and colors molecule)

iv. Medical

- Antibiotic
- Painkiller
- Stimulant
- Gender
- Anti-anxiety
- Anti-ulser
- Anti-viral
- Anti-cancer
- Anti-AIDS

However, only certain molecules data are used while performing the tasks in this research. The data used are mostly downloaded from Molecular Database sites and are in .pdb file format. The source URL of these sites is:

http://www.biochem.umd.edu/biochem/kahn/teach_res/prtn_DNA_tut/

http://www.ch.ic.ac.uk/vchemlib/mol/mol.html

Overall, the molecular data that have been used are:

- 1aay.pdb – DNA molecule data
- 1HVR.pdb – protease enzyme molecule data
- 1IGM.pdb – protein data
- 2JEL.pdb – protein data
- alkane.zmf (trajectories) – trajectory molecule data which consist of multiple time-steps (2000 steps in Butana, 2000 steps in Pentana and 2000 steps in Hexana)

- but_cluster_3_1.idx – this subtrajectory of alkane molecule belongs to metastable conformation of butane (time step from 1-798)
- h2o.pdb – water molecule
- methyl_methacrylate.pdb – polymer molecule of type Acrylates

5.2 Description on Modules Involved

5.2.1 Align Molecules

This module aligns two arbitrary molecules to each other. However, the molecules need to fulfill one precondition: they both must have (at least) one level, the align level, with the same name and an equal number of groups. Two molecules are required to do alignment.

MoleculeA [required]

- First molecule to which the second molecule will be aligned.

MoleculeB [required]

- Second molecule to be aligned to the first one.

Ports:

Options

- multiple transforms: If this toggle is not selected, a local minimum will be found starting from the current positions of the molecules. If the toggle is selected, molecule B will be translated such that the centers of the molecules according to the groups found in the Align Level are at the same position. Starting from this position, 20 rotations will be applied to molecule B, and these new positions will be used as starting points for the alignment.

- show alignment: If this option is set, user can observe the alignment process visually. Otherwise, only the final alignment will be shown.

5.2.2 Align Sequence

This module aligns the sequences of two molecules to each other. Proteins as well as nucleic acids can be aligned. Currently, it is not possible to align t-RNA sequences because they contain modified bases deviating from the standard data format. The module will be adjusted and extended in the near future. The output of the alignment is displayed in a separate window. Currently user may choose between *local*, *semiglobal*, and *global* alignment. The algorithms are based on the Waterman algorithms for pairwise sequence alignments using a *Blosum* weighting matrix for protein sequences and a *Transition/Transversion* weighting matrix for nucleic acids. Consider the following three cases to decide which algorithm to use:

- *local alignment*

The local alignment algorithm is the best algorithm to use user want to match a short sequence, possibly a subsequence or motif of some molecule, against a long one. The local alignment algorithm is not useful for aligning two long sequences to each other, e.g., two related proteins. For alignment of two long sequences, use one of the next two algorithms instead.

- *global alignment with gap function*

In this case, it matches two related long sequences to each other, e.g., two related enzymes. The algorithm matches the whole sequences penalizing long gaps proportionally less than many short gaps.

- *semiglobal alignment*

If user have no idea about the relationship of 2 test molecules, it is best to use the semiglobal algorithm first because it will scan for the possibly best matching parts of molecules without having to align the whole sequences as with the global algorithm.

To perform sequence alignment, the following connection is required:

MoleculeA [required]

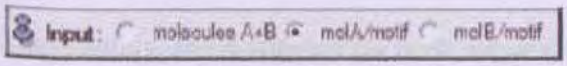
- First molecule to be aligned.

MoleculeB [required]

- Second molecule to be aligned.

Ports:

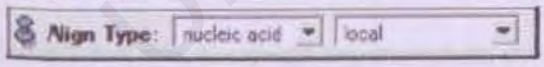
Input



Input port defines which sequences should be taken as input for the alignment. Three options exist:

- molecule A and B.
- molecule A and the specified motif.
- molecule B and the motif.

Align Type



In order to align proteins as well as ribonucleic acids, user needs to specify the type of molecule in the first drop down menu. In the second menu user can select one of three algorithms as explained above: *local*, *semiglobal*, or *global* alignment.

Once aligned, the output should be a pop-up window (as shown) displaying the sequences.

AlignView

Figure 5.1: Pop-up window displaying the sequences between two aligned molecules

To provide an idea about where in the sequences similarities have been found, the rest of the sequence is displayed as well (in red). To find out where in the sequence user are, just click at the position within the sequence user are interested in.

Immediately, the positions for both sequences will be shown in the lower-left corner of the window. The clicked position will be marked by a black rectangle.

5.2.3 Bond Angle View

The *BondAngleView* offers an alternative to the *MoleculeView*. This viewing module, however, does not display atoms and bonds but the bond angles, i.e., for every three atoms connected by two bonds a triangle will be shown. The vertex colors are linearly interpolated across the triangle. Apart from coloring the molecule as is done in the MoleculeView, the bond angles can be colored according to a specified color field, e.g., the electrostatic potential, and a colormap.

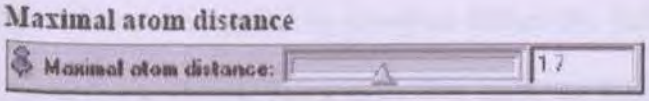
5.2.4 Bond Calculation

This module calculates bonds in a given molecular structure by analyzing the distances between the individual atoms of the structure. A maximal distance value will be used to determine whether two atoms are connected or not.

This module is especially useful when being used with a molecule that is derived from a trajectory.

Each newly displayed time step will cause the bonds to be recalculated. A similar module for bond calculation is available via the molecule editor.

Ports:



- Maximal atom distance

The value determines the maximal distance between two atoms (in ° A) which they are considered to be connected by a bond.

5.2.5 RankTime Step

This module takes an object of type MolTrajectory as input and either sorts all molecules in the trajectory or selects a single molecule from the trajectory. Selection and sorting can either be done using the *observable* values of the trajectory or the rmsd (root mean square deviation) value in comparison to a specific molecule, which then needs to be connected to the module.

The module is very helpful in conjunction with the module for the computation of the mean molecule. It allows user to find the time step in the trajectory that best approximates the mean molecule.

5.2.6 CompMolSurface

This module computes molecular surfaces. Three types of molecular surfaces can be generated: the *van der Waals (vdW) surface*, the *solvent accessible surface (sas)*, and the *solvent excluded surface (ses)*. The *vdW* surface encloses all van der Waals spheres of the molecule's atoms. The *sas* encloses all van derWaals spheres extended by some

probe radius. Finally, the *ses* encloses the subspace which is not accessible to a probe sphere in the presence of the molecule represented by its van der Waals spheres. [32]

The implemented algorithm allows the computation of the full molecular surface as well as of partial surfaces. If the partial surface option is selected, the surface will only be computed for all highlighted atoms.

Apart from the option to compute either the full or a partial surface, user can also choose between two algorithms that differ in the quality of the surface generated, and also in speed. If time does not matter, user should always use the default option, *correct*. However, if user is interested in the dynamic behavior of the molecular surface, time does matter and user might want to use the second algorithm.

As a result of the computation a new object, the molecular surface, will appear in the object pool.

To visualize the surface, attach the MolSurfaceView module to it.

5.2.7 Configuration Density

This module enables user to compute a probability density for the positions of atoms and bonds within a molecular trajectory. The input is an object of type MolTrajectory. As output, either a scalar field or a color field is generated. In order to visualize the computed density, user can apply the volume rendering module, Voltex.

Since the time steps of the molecular trajectory can be arbitrarily rotated and translated, user must perform an alignment for each time step to fit it best to some chosen reference. This is done internally, but user must specify how the molecules should be aligned to each other. There are four options to choose from. The first uses all atoms for

alignment, which is the recommended option. In the second option, user can select a few atoms. The third and fourth use *none* and *center of gravity* alignment, respectively.

For the representation of the molecule two kinds of geometric objects can be used: spheres and sticks.

Here, spheres represent the positions of the atomic nuclei and the sticks the existing bonds within the molecule. The following connections are required in order to compute configuration density:

Data [required]

- The molecular trajectory for which the density should be computed.

AlignMaster [optional]

- The molecule to which each time step of the trajectory will be aligned.

PrecomputedAlignment [optional]

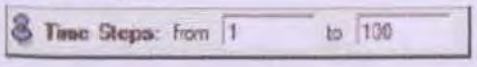
- Instead of aligning all time steps of a trajectory to the *AlignMaster* molecule, user can use a precomputed alignment to align the time steps.

Continuous and DiscreteColormap [optional]

- These two colormaps are used for the color management of the computed volume, in case a color field should be generated.

Ports:

Time Steps



This port specifies range of time steps for which the density should be computed.

Voxel Size

This port determines the size of a voxel, of the field storing the probability density.

5.2.8 MeanMolecule

This module computes the mean molecule of a molecular trajectory by averaging the atom positions. In order to do this, the molecules need to be transformed to a common coordinate system. This can be done either by aligning all time steps to some reference molecule, or by computing transformations with the PrecomputeAlignment module. The following connections are required:

Data [required]

- Molecular trajectory for which the mean molecule should be computed.

AlignMaster [optional]

- In order to compute the mean molecule, all molecules of the trajectory need to be aligned with respect to a certain molecule. This is done by right-clicking on the little rectangle of the module icon in the object pool, selecting *AlignMolecule*, and connecting it to a molecule in the object pool.

5.2.9 Measurement

This module lets user determine distances, angles, and dihedral angles between atoms of the molecule. To obtain a measurement, user need to select the atoms that are to be examined. In order to select objects in the viewer, user needs to switch into the interaction mode of the viewer.

5.2.10 MolSurfaceView

This module visualizes data objects of type MolSurface. Objects of type *MolSurface* can be either molecular surfaces, such as *van der Waals, solvent accessible,* or *solvent excluded surfaces*, generated by the CompMolSurface module.

5.2.11 MoleculeLabel

This module allows user to label certain parts of the molecule. If a *MoleculeLabel* module is connected to an object of type *Molecule*, all clicks in the viewing window normally handled by a viewing module connected to the same molecule will instead be handled by the *MoleculeLabel* module, by default. If the selection mode is *atoms*, a label for the selected atom will be displayed; if it is *residues*, a label for the residue the atom belongs to will be displayed. Up to two data items per label can be displayed.

5.2.12 MoleculeView

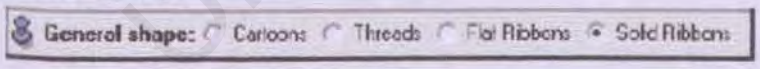
The *MoleculeView* allows user to display molecules in three different representations: atom spheres, wire frame, and ball-and-stick. For each of these representations there are two *Quality* modes, *fast* and *correct*.

5.2.13 SecStructureView

This module provides the functionality for viewing the secondary structure of a molecule. It relies on the information given in the data entry of the connected molecule. Helices, sheets, and turns can only be displayed if information about these structures has been read in (for example from a pdb file).

Ports:

General Shape



This port determines the general shape for viewing the secondary structure. The following view modes are available:

- Cartoons: In the *cartoon* mode, user can select a view mode for each secondary structure type separately. The backbone and the turns will be shown as tubes that run through the central atoms of each residue.

- Threads: The backbone will be drawn as a set of lines running parallel to the peptide planes of the amino acids. This mode will show the surface orientation of the backbone, thus making parts containing symmetries (like helices or sheets) easily recognizable. It is also the view mode which can be rendered fastest.
- Flat Ribbons: Same as the thread mode, but the areas between the lines will be a filled surface, thus giving a better surface impression with the trade-off of slightly lower rendering performance.
- Solid Ribbons: Unlike flat ribbons, solid ribbons will have an elliptic cross section.

5.2.14 TubeView

With the *TubeView* module, user can connect an arbitrary set of consecutive atoms with a tube through interpolated atom coordinates.

5.2.15 Molecule Editor

This tool can be used to change the geometry and topology of a Molecule data object. To change group attributes, user can use the Molecule Attribute Editor.

Most of the editor's functions are applied to sets of atoms, which can be selected by using the selection browser or by directly clicking on atoms in the viewer.

The molecule editor has two tabs:

i. The *Transform* tab contains all tools to change the geometry of selected atoms by adjusting positions, bond lengths, torsional angles, and bond angles.

ii. The *Tools* tab offers methods to split or copy parts of the molecule, as well as an interface for adding or removing bonds between atoms.

- Transform Tab

The transform tab is divided into four different sections. Each section can be used to adjust certain coordinates of the currently selected atoms. The different coordinate types are:

i. *Position* is the Cartesian coordinate of an atom. If several atoms are selected, only relative changes are allowed.

ii. *Bond Length* is the distance between two selected atoms.

iii. *Bond Angle* is the planar angle between three selected atoms.

iv. *Bond Torsion* is the dihedral angle between four atoms.

- Tool Tab

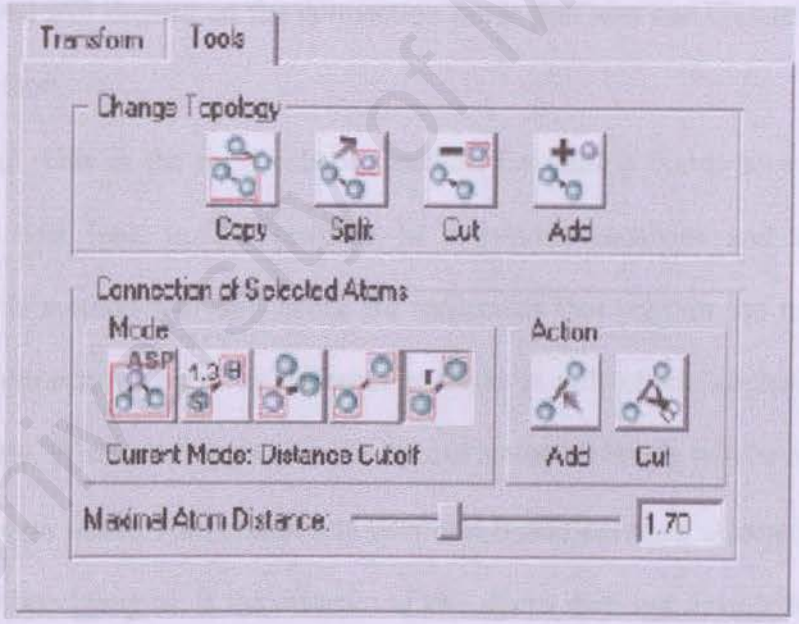


Figure 5.2: Tool tab

The tools tab is subdivided into two sections:

i. Change Topology

The *cut* button will cut the currently selected groups out of the molecule.

The *split* button will do the same but will copy the groups into a new molecule which will be added in the object pool.

The *copy* button will leave the current molecule unchanged while copying the selected groups into a new molecule which will be added to the object pool.

When pressing the *add* button, a window will open which will let user choose another molecule in the object pool whose groups they want to add to the current molecule.

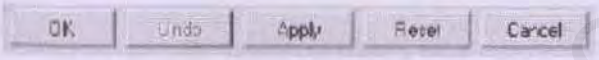
ii. Connection

The *Connection* section offers different options for influencing the bonding of the currently selected atoms. On the right side of the interface are buttons for adding or removing bonds between the currently selected atoms. The set of bonds which will be added or removed will depend on the connection mode that user can choose on the left side of the interface.

- *Standard*: This is the most reliable method for adding bonds to a protein or DNA/RNA. It will look up all residues in a residue database and add bonds accordingly. This method will only work for molecules that contain the residue type attribute. For connections between different residues it will check all residues on a chain sequentially. When used together with the cut action all bonds will be removed.
- *Bond length table*: This option lets user add bonds between selected atoms by looking up their bond lengths. If the distance of two atoms does not deviate further than a certain threshold from the bond length between the respective elements in the table, the bond will be added. This method is able to distinguish between single, double, triple, and aromatic bonds.

- *External*: This mode can only be used together with the *cut* action. It will remove all bonds between selected and unselected atoms thus enabling user to disconnect certain parts of the molecule from the rest.
- *All*: This mode will add or remove all possible bonds between the selected atoms.
- *Distance Cutoff*: This last mode uses a distance cutoff for deciding which bonds to add or remove. The *Maximal Atom Distance* slider determines the maximal distance of two bonded atoms in ° A. When using this mode together with the *add* action all bonds between atoms which are nearer than the cutoff distance will be added. Equally, the *cut* action will remove all bonds whose length is greater than the threshold.

Button Group



The *OK* button will close the editor and accept all changes made to the molecule.

The *Cancel* button lets user return from the editor restoring the original state.

Resetting the molecule without canceling the editor can be done with the *Reset* button.

The *Apply* button will apply the changes you have made to the molecule object. This means that any further reset will return the molecule to the current state.

With the *Undo* button any transformation can be undone up to 10 levels back from the current state.

5.3 Rendering, Segmenting, and Visualization Methods Manual

5.3.1 Load Data into the System

Molecule data that are used in this task: 1aay.pdb

Several molecular file formats including, for example, PDB, Tripos, and UniChem, can be read and written by amiraMol, other file formats, such as CHARMM, can only be read.

To load data into Object Pool,

File-> Load->XXX.pdb.

A green icon will appear in the object pool. The green icon represents the object of type Molecule. Click on the green icon and the working area will display the information regarding the molecule, such as the number of atoms, etc.

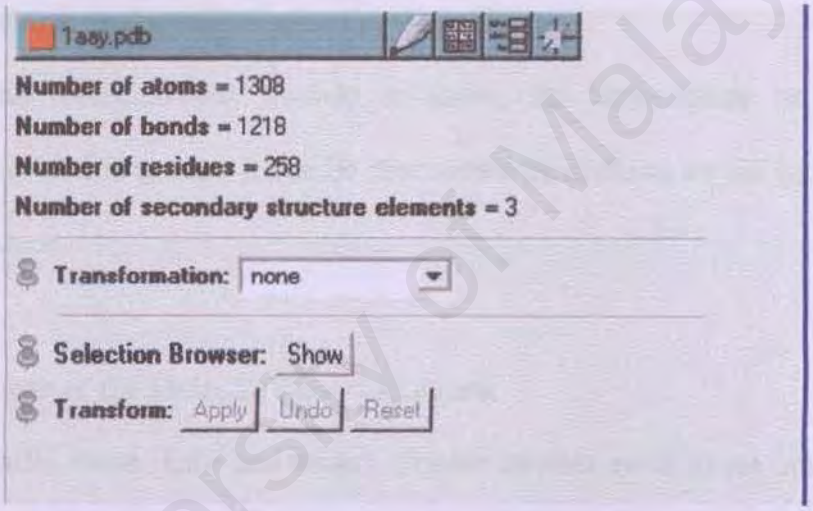


Figure 5.3: The working area displaying the information regarding 1aay.pdb molecule

5.3.2 Rendering

Molecule data that are used in this task: 1aay.pdb

Displaying the Molecule with the MoleculeView Module

The MoleculeView module is the basic display module for visualizing molecules. It allows user to display atoms as plates or balls and bonds as lines or cylinders.

To render a molecule,

Click on the green data icon in the object pool using the right mouse button->

A menu, containing several entries and submenus, will appear.

->Select the entry MoleculeView.

A new yellow icon labeled MoleculeView appears in the object pool. Yellow icons, in general, represent display modules, i.e., modules that visualize objects in the viewer. The blue line between the icons indicates a connection between the objects.

-> go to viewing mode, in which case the mouse cursor is displayed as a hand. This enables to manipulate the display of molecule in viewer.

Whenever the MoleculeView module is active, the little square on the yellow MoleculeView icon is orange. It can be deactivated by clicking on the square with the left mouse button.

Some basic ports of the MoleculeView module are:

i. Mode port (balls, sticks, balls and sticks): Choose another mode to see both atoms and bonds of the molecule, or just atoms. If atoms are shown, use the Atom Radius port to adjust the size of the atoms as desired.

ii. Quality port (fast, correct): Use the fast mode to display a large molecule. If the graphics hardware is fast enough, the correct mode can be used even for large molecules.

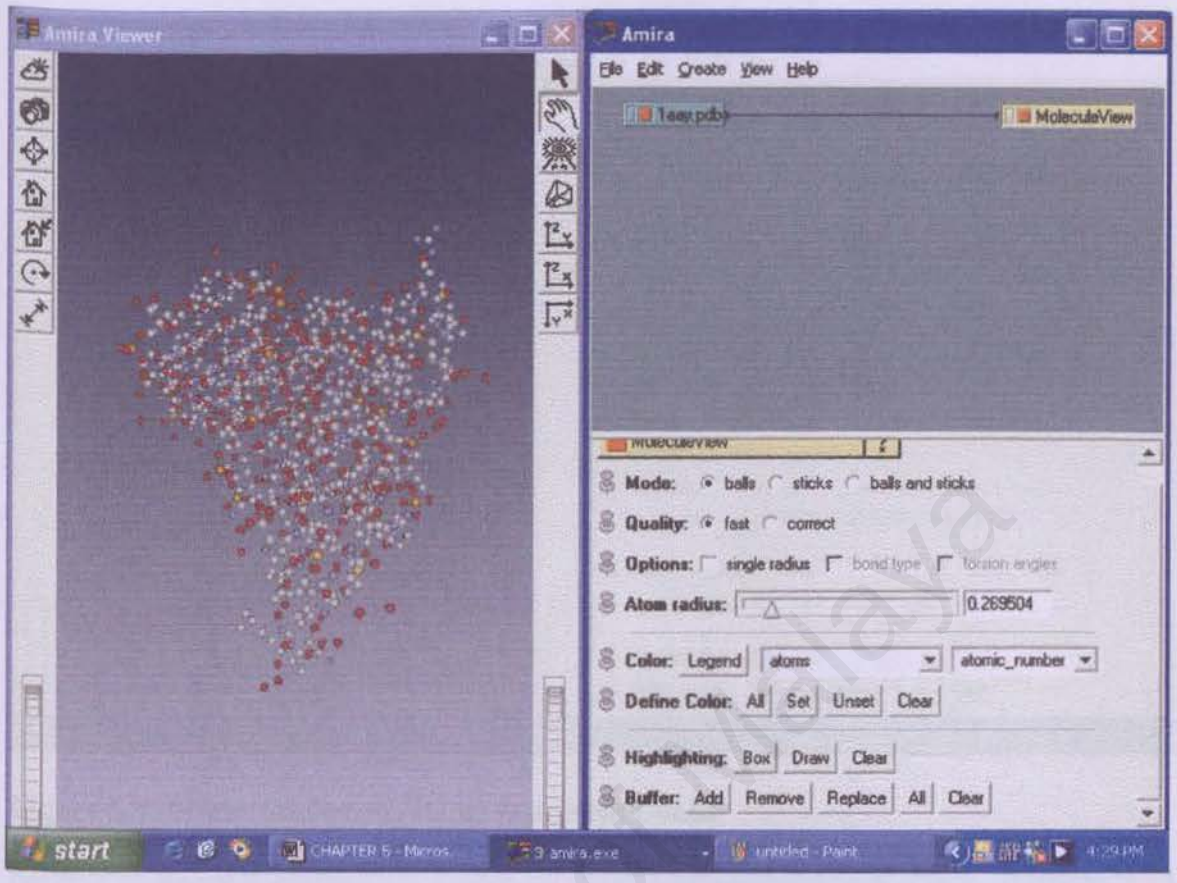


Figure 5.4: Balls rendering, balls represent the atoms

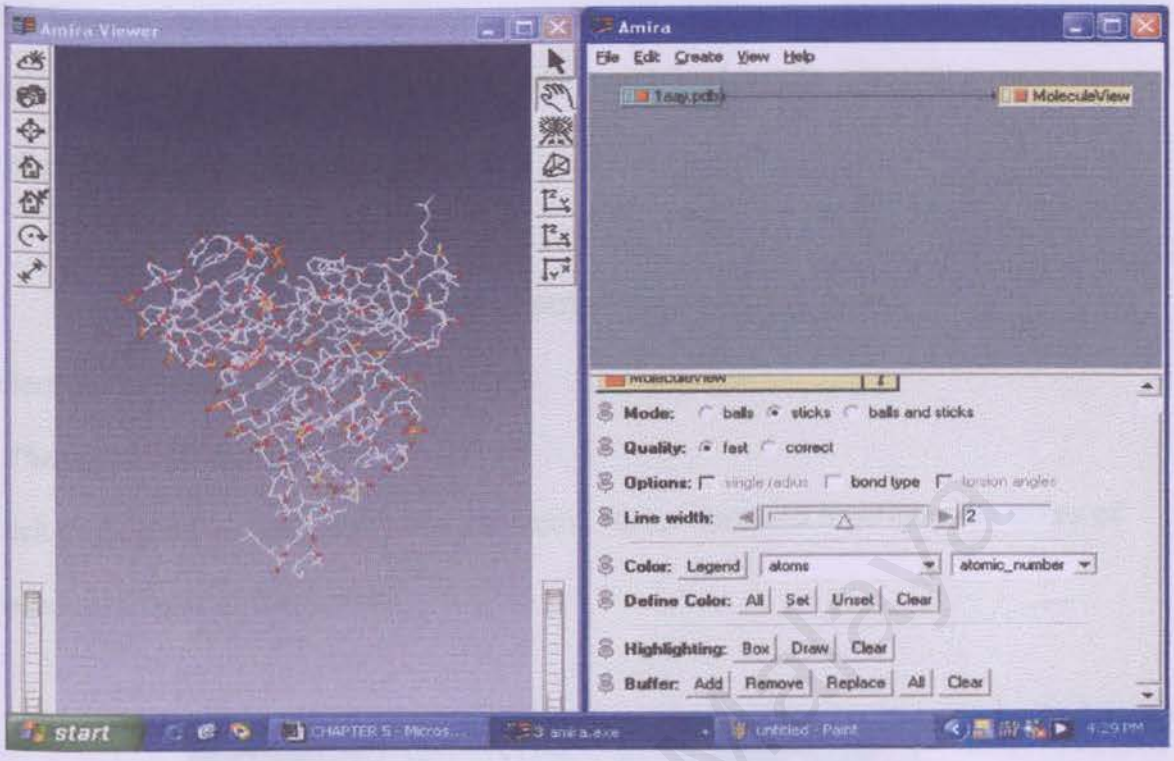


Figure 5.5: Sticks rendering, sticks represent bonds

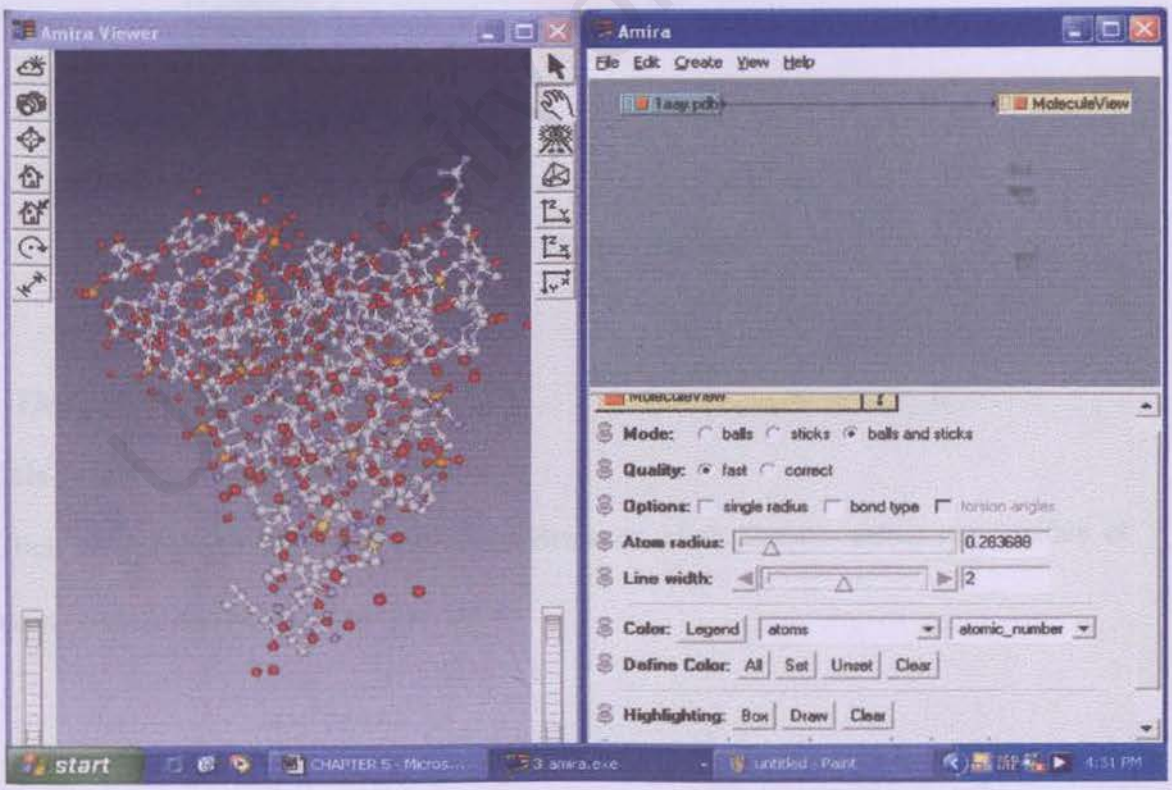


Figure 5.6: Sticks and balls rendering

5.3.3 Segmenting

Molecule data that are used in this task: methyl_methacrylate.pdb

In segmenting, 2 tasks are to be performed:

1. Selecting atom/residues of interest.
2. Change color, remove the selected part or restore back, labeling.

Selecting molecules can be done in 4 ways:

i. Through Color Scheme

Click on Legend-> click on any options below Name. This will highlights the atoms of the selected color in the viewer.

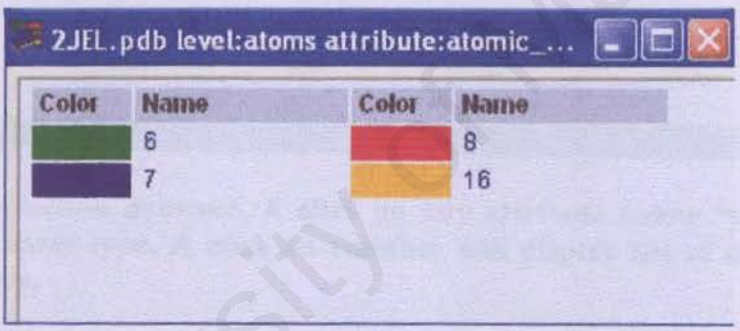


Figure 5.7: Color Legend

ii. Through Selection Browser

This enables a more precise selection.

Click on the green icon-> on Selection Browser, click Show-> select any residue or atom -> click on the atom/residue name.

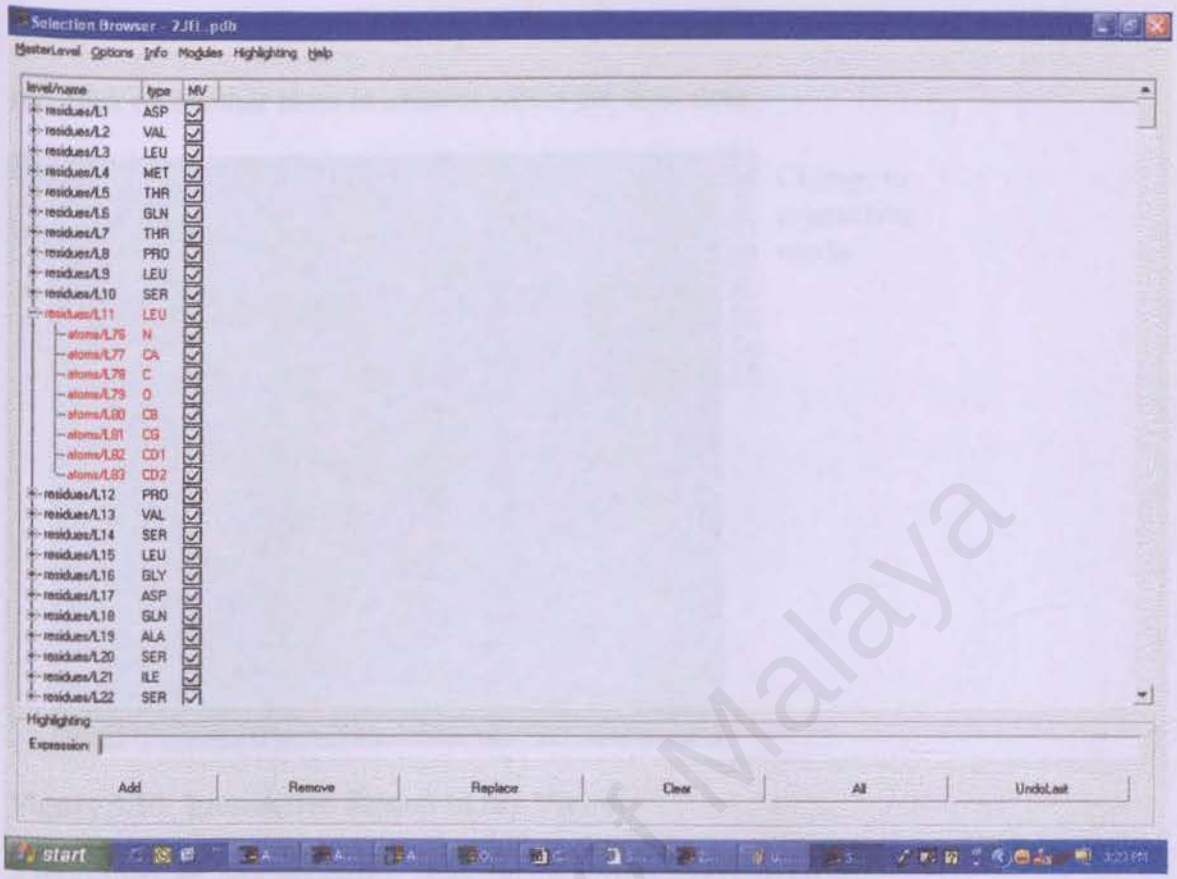


Figure 5.8: Selection Browser. A click on any attribute below "type" will select atoms of the same type. A click on residues will display list of atoms below the selected residue.

iii. Using Draw Tool

Click back to the yellow icon-> on Highlighting, click Draw-> go to viewer and keep pressing left button while moving the mouse on selected parts of the molecule.

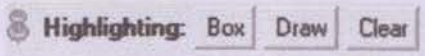


Figure 5.9: Draw Tool

iv. Interactive Selection

This enable user to select molecule directly from the viewer.

Change the viewer into interaction mode-> click on any atoms-> hold on Shift button and click on another atom in order to select the next atom.

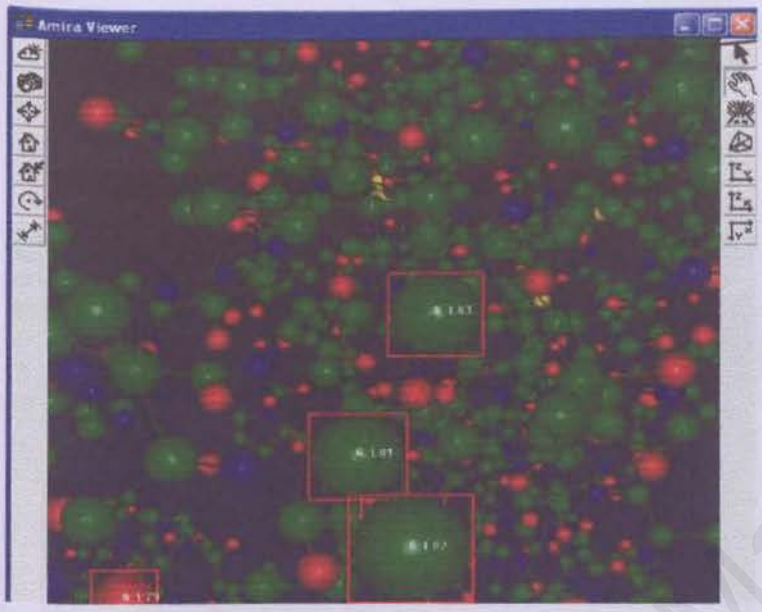


Figure 5.10: Interactive Button in the Viewer

After selection, one can define color on the selected part, to do so,

Go to Define Color-> Click Set-> Color Dialog box will pop up-> Select appropriate color scheme (RGB/HSV), RGB represents Red/ Green/ Blue, meanwhile HSV, also known as HSL, stands for Hue/ Saturation/ Lightness-> click Apply.

To restore the original color, click Clear.

To remove the selected molecule, there are four function buttons that perform different actions:

Go to Buffer, click on

1. remove- will remove the selected parts
2. replace- will remove all other parts other than the selected parts
3. display- will display/restore the whole molecule
4. clear- will clear the whole molecule

To remove the highlighting box, go to Highlighting->click Clear.

In labeling, right click on the green icon-> Display-> MoleculeLabel-> in working area, select appropriate options in labeling. To label the atom of interest, go to Labels-> click Add Highlighted Groups. Click Clear to unlabel.

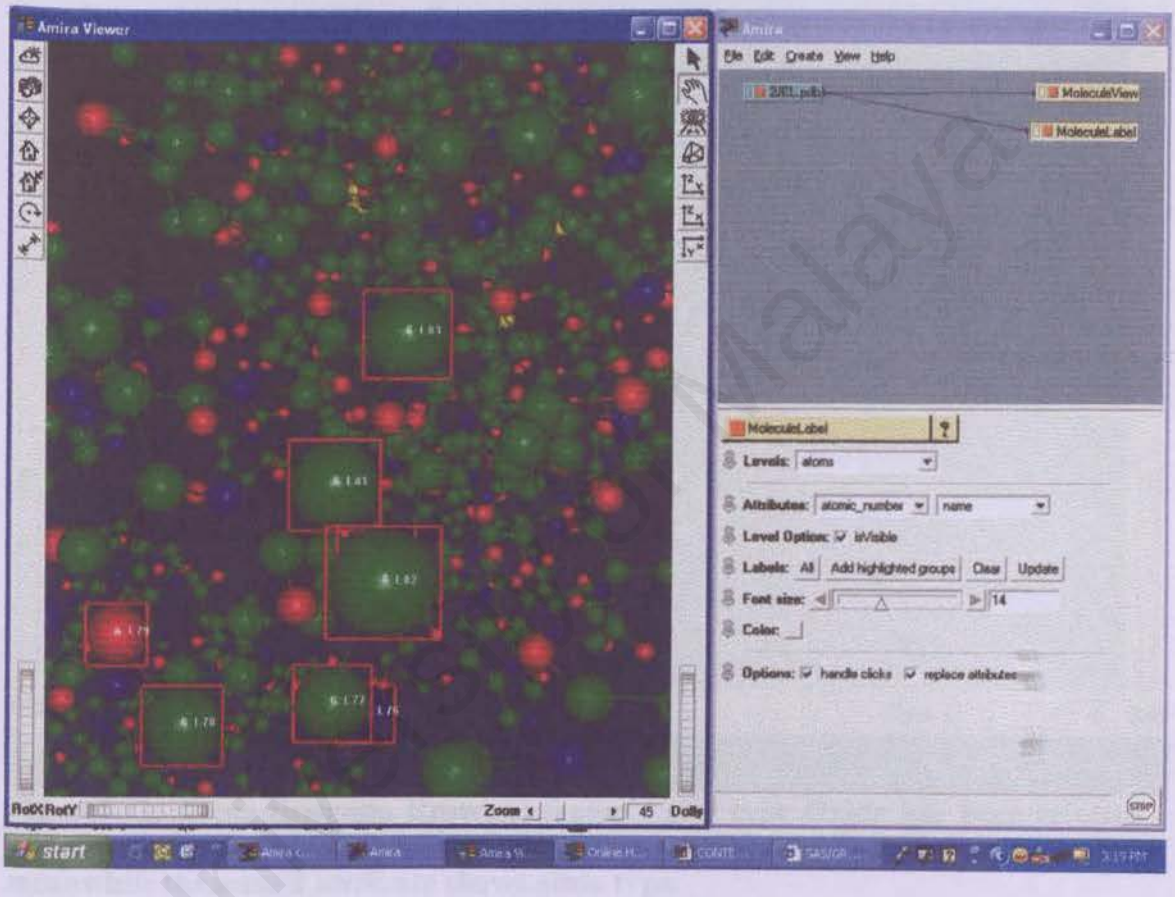


Figure 5.11: Atoms of interest that have been labeled. The first column shows the atom number and the second column shows atom name.

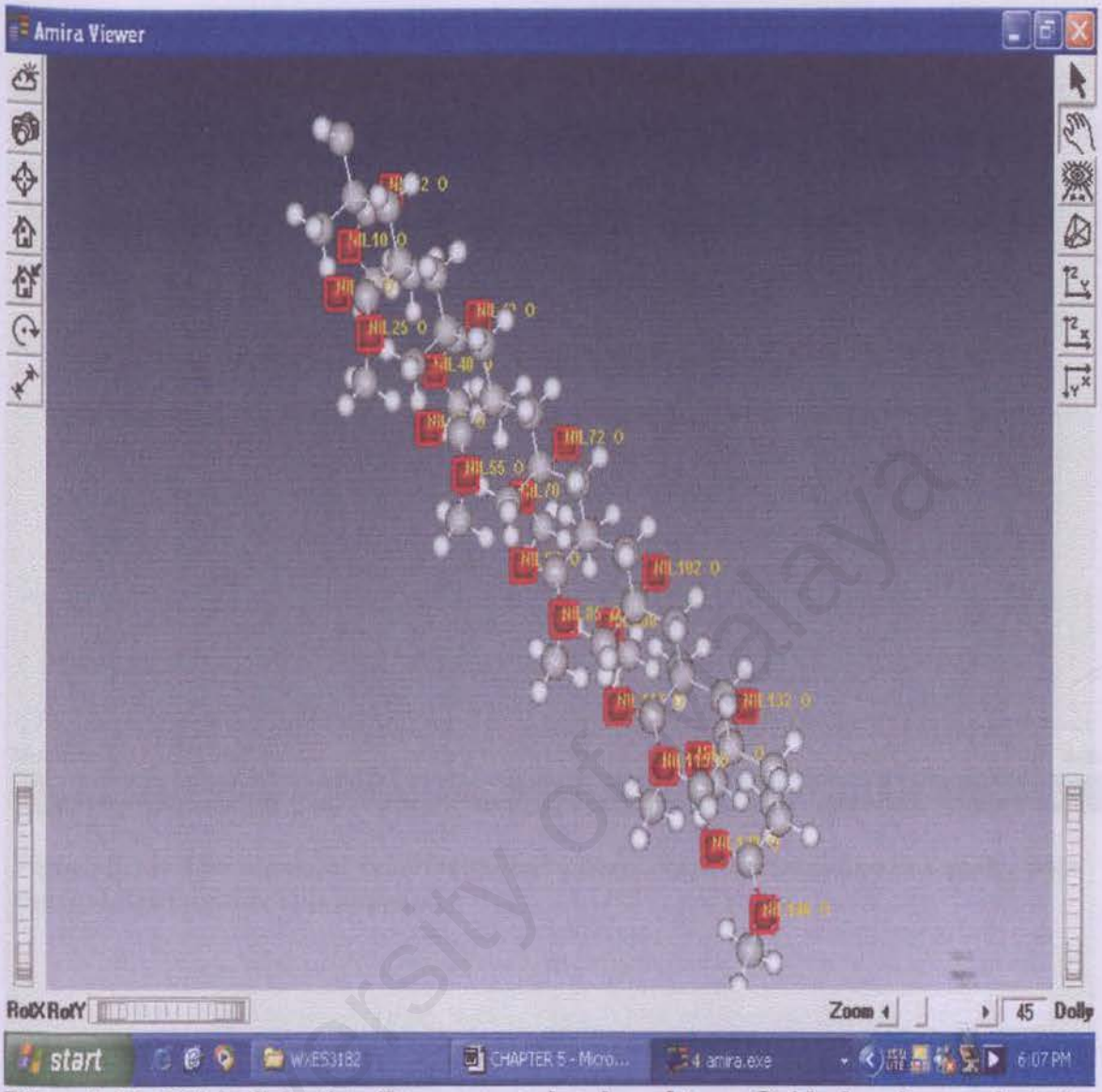


Figure5.12: Using Selection Browser, molecules of type Oxide has been selected and labeled using MoleculeLabel, the first attribute shows the atom name, meanwhile the second attribute shows atom type.

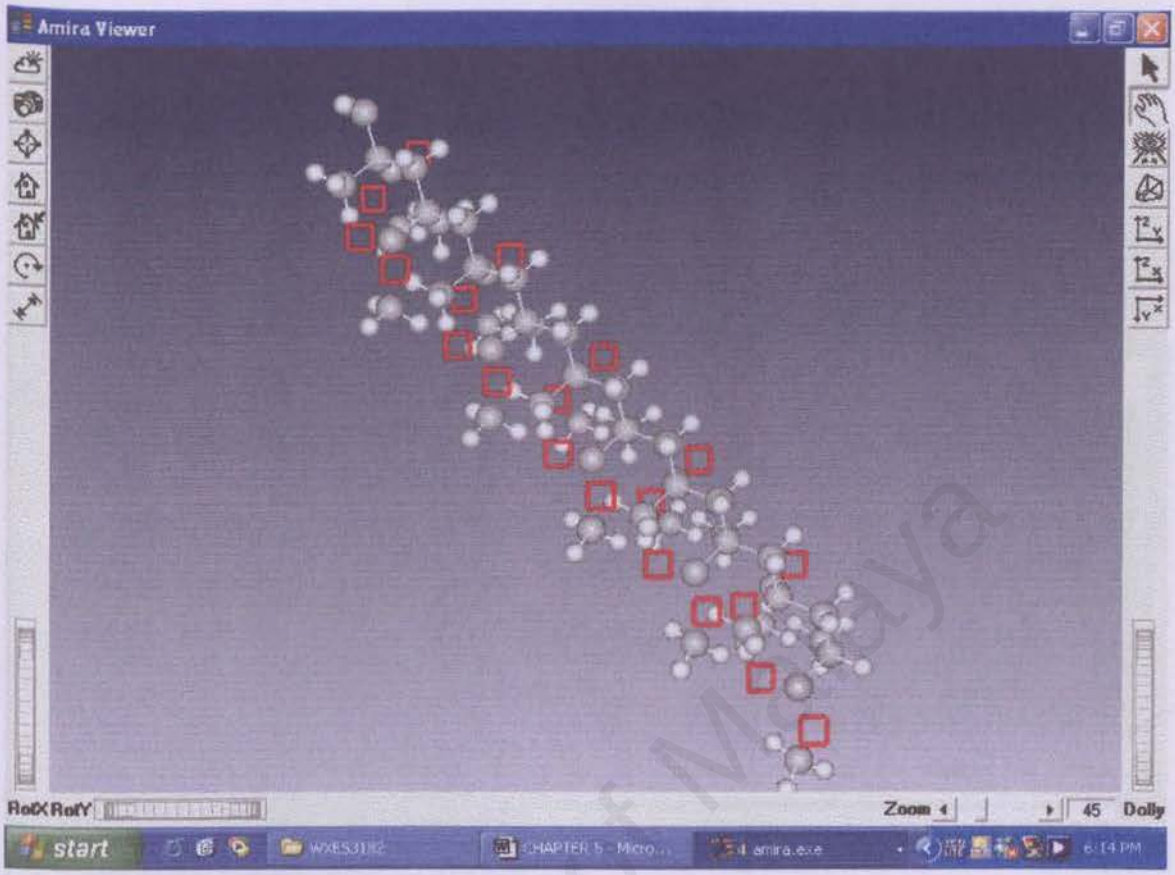


Figure 5.13: The atoms of type Oxide have been removed from the molecule, with the highlighting box still appears

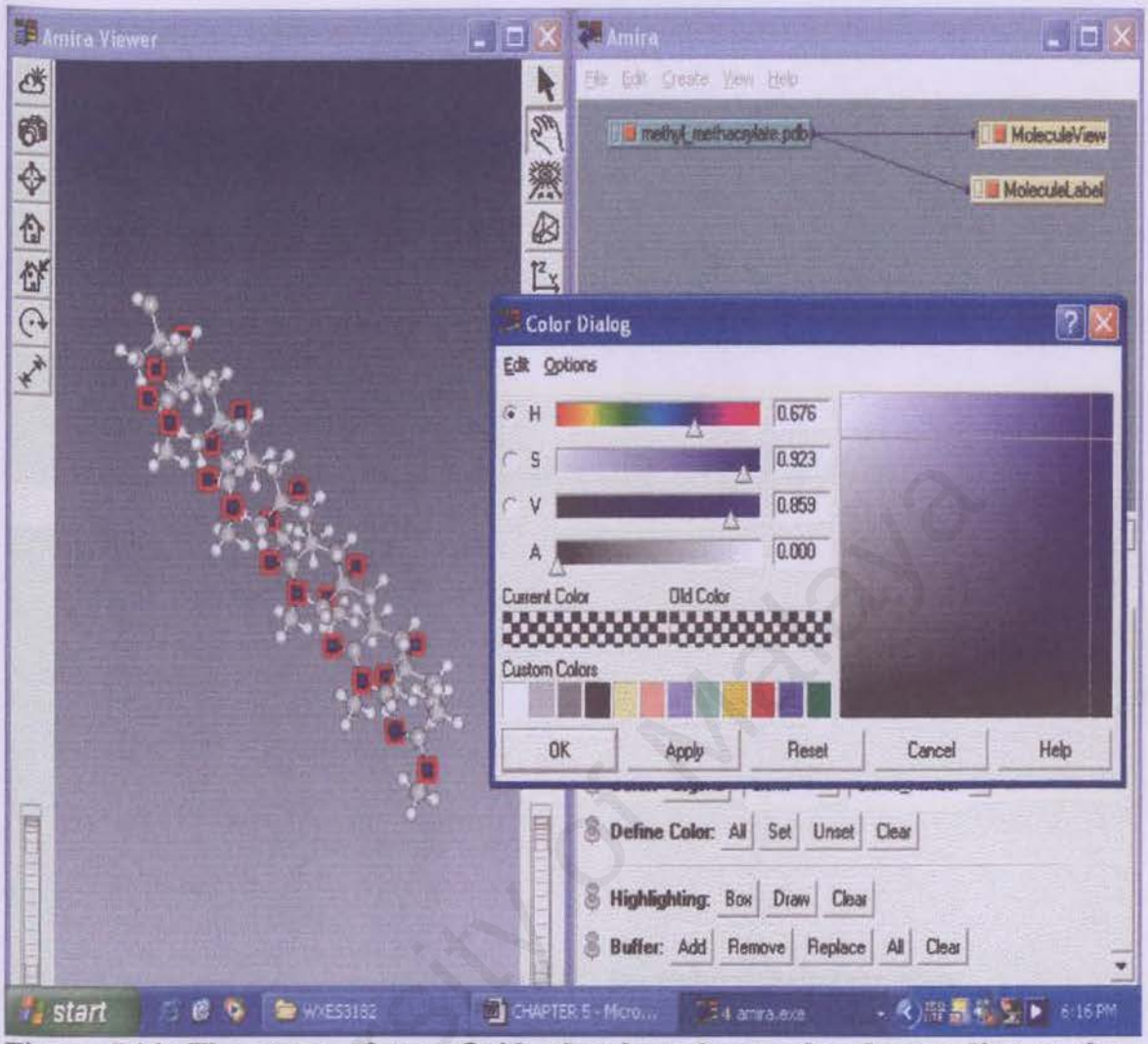


Figure 5.14: The atoms of type Oxide that have been colored according to the selected color as specified by user in the Color Dialog

5.3.4 Visualization

Molecules can be visualized with the Molecule View, Bond Angle View, SecStructureView and TubeView. A combination of different views is allowed in Amira. Besides, user can also select which atom/residues of interest to be visualized in different views. This can be done through Selection Browser.

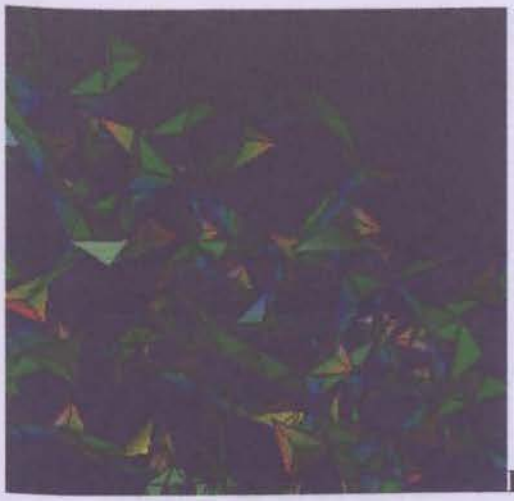


Figure 5.15: Bond Angle View

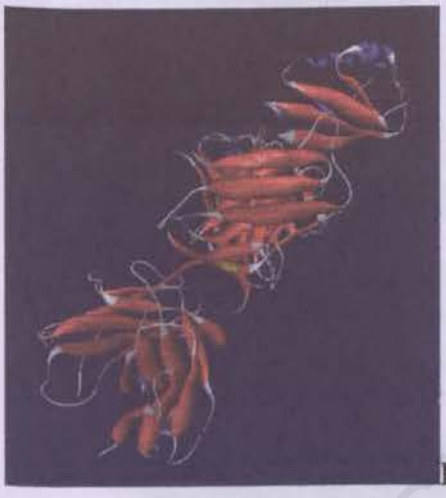


Figure 5.16: Secondary Structure View

Figure 5.17: Tube View

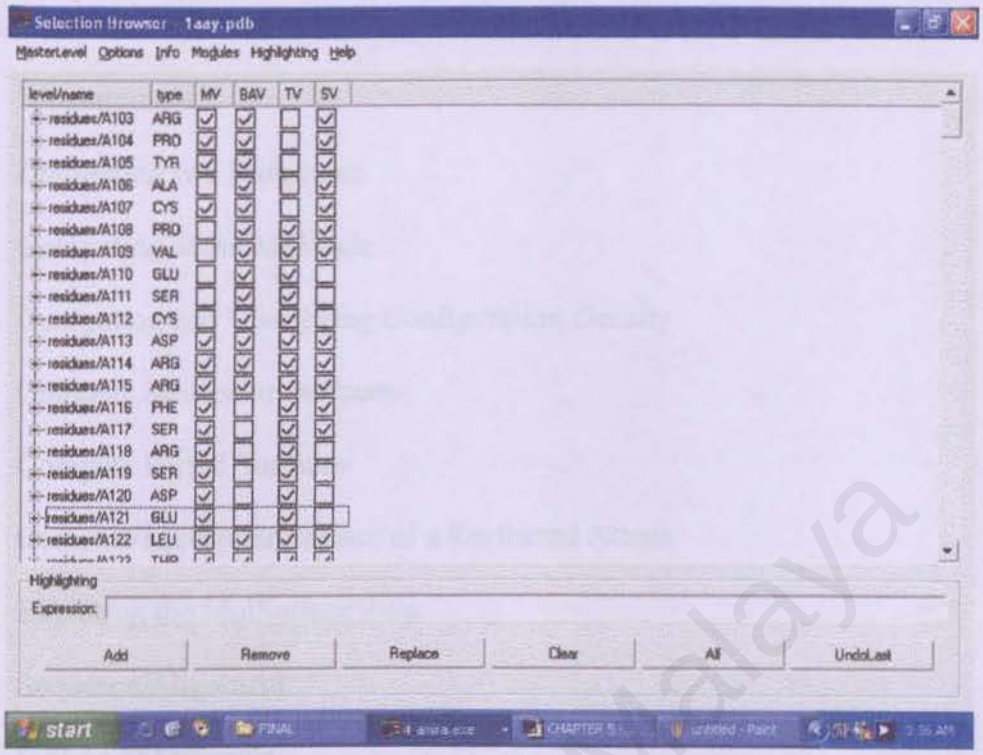


Figure 5.18: User can combine different type of views in any residues of a molecule by checking (tick) at boxes below MV (molecule view), BAV (bond angle view), TV (tube view) or SV (secondary structure view)

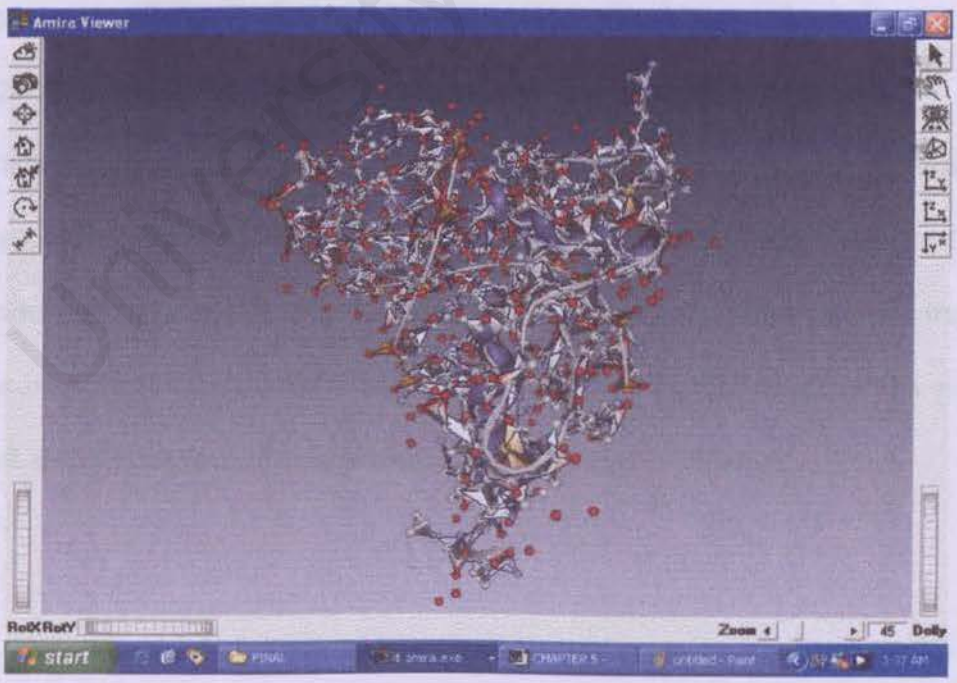


Figure 5.19: The combination of different views for 1aay.pdb molecule

The tasks that are to be performed in visualization include:

1. Measurement
2. Comparing two Molecules
3. Computing Mean Molecule
4. Computing and Visualizing Configuration Density
5. Compute Molecular Surfaces
6. Compute Partial Surfaces
7. Compute Molecular Surface of a Restricted Atoms
8. Exploring the MolSurfaceView
9. Sequence Alignment
10. Editing of Molecule

Task 1 are much likely performed using Measurement Module of AmiraMol. Task 2-4 involved the Align Molecules Module. Task 6-9 involved Molecular Surface Module of AmiraMol and lastly, task 10 requires Molecule Editor Module.

Task 2-4 are related to visualization of metastable conformations. The phrase "metastable conformation" indicates a dynamic aspect of molecular behaviour; it denotes metastable shapes, for example molecular geometries which survive the fast oscillations around equilibrium positions, this means configurations whose trajectory remains inside a set for a long time period before leaving it eventually. To visualize metastable conformations, the following road map is to be followed:

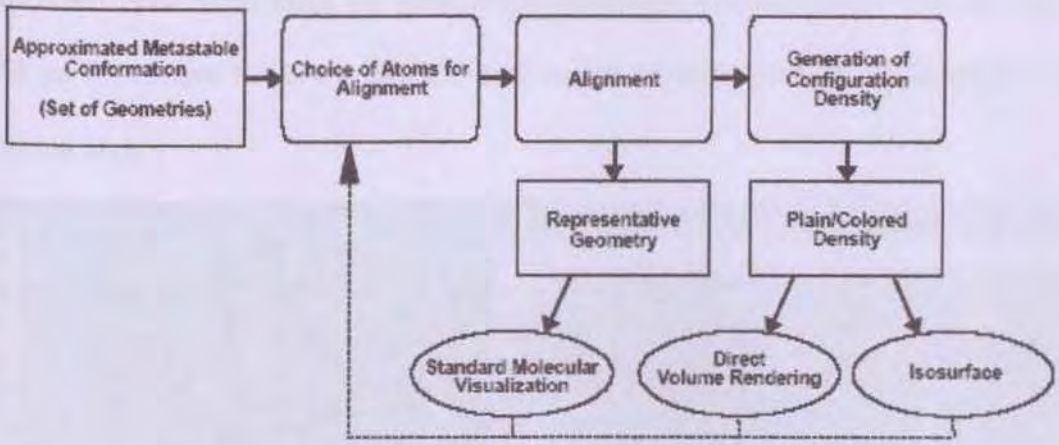


Figure 5.20: Road map for the visualization of metastable conformation

Alignment is necessary to eliminate the differences caused by global rigid transformations. After computation of alignment, two approaches are available. First, is to identify a single geometry that can be considered as a representative and to depict it using standard molecular visualization techniques. The second approach aims at focusing on the geometric variability of the metastable conformation. For this, the configuration density is computed. This density can be visualized using direct volume rendering (voltex) or isosurfaces. [31]

1. Measurement

Purpose: To measure distance between 2 atoms and to measure angles in between 3 or 4 atoms.

Molecule data that are used in this task: 1aay.pdb

On Object Pool, right click on MoleculeView->click Measurement-> On the viewer, click on Interactive Mode->select 2, 3 or 4 atoms. Information will be displayed at the working area.

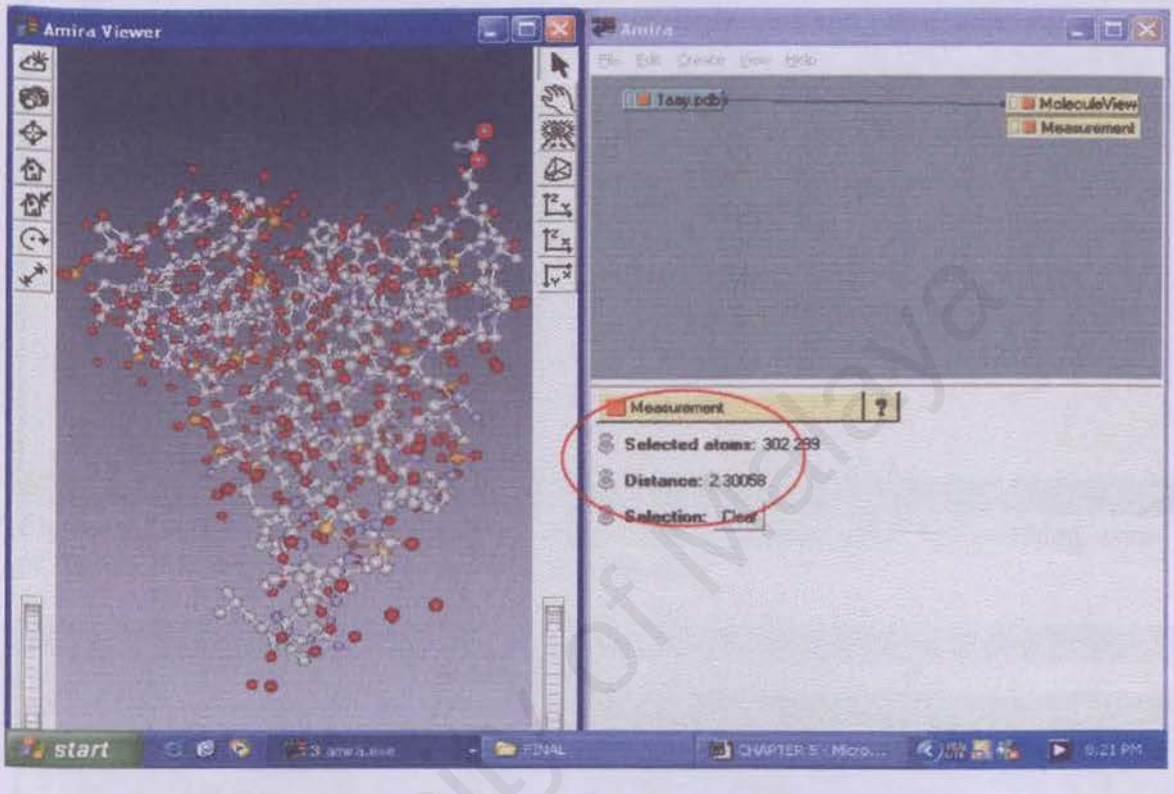


Figure 5.21: The distance between two atoms will be displayed at the working area of Amira if two atoms are selected at the viewer

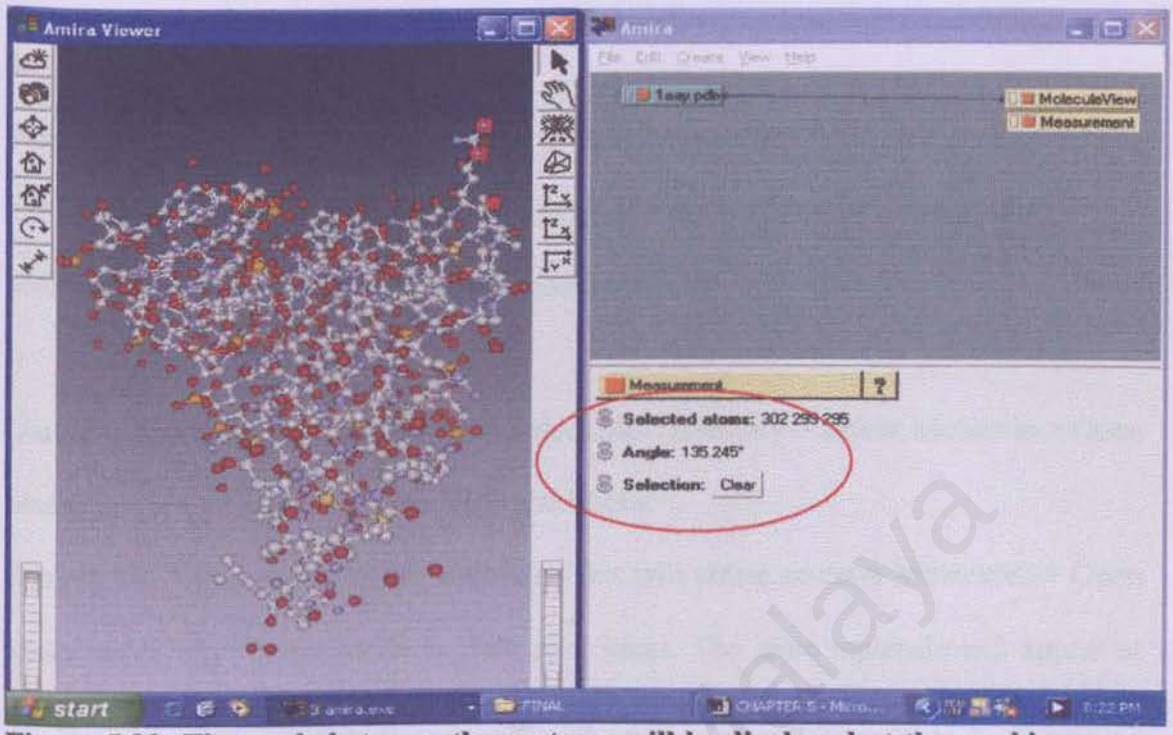


Figure 5.22: The angle between three atoms will be displayed at the working area of Amira if three atoms are selected at the viewer

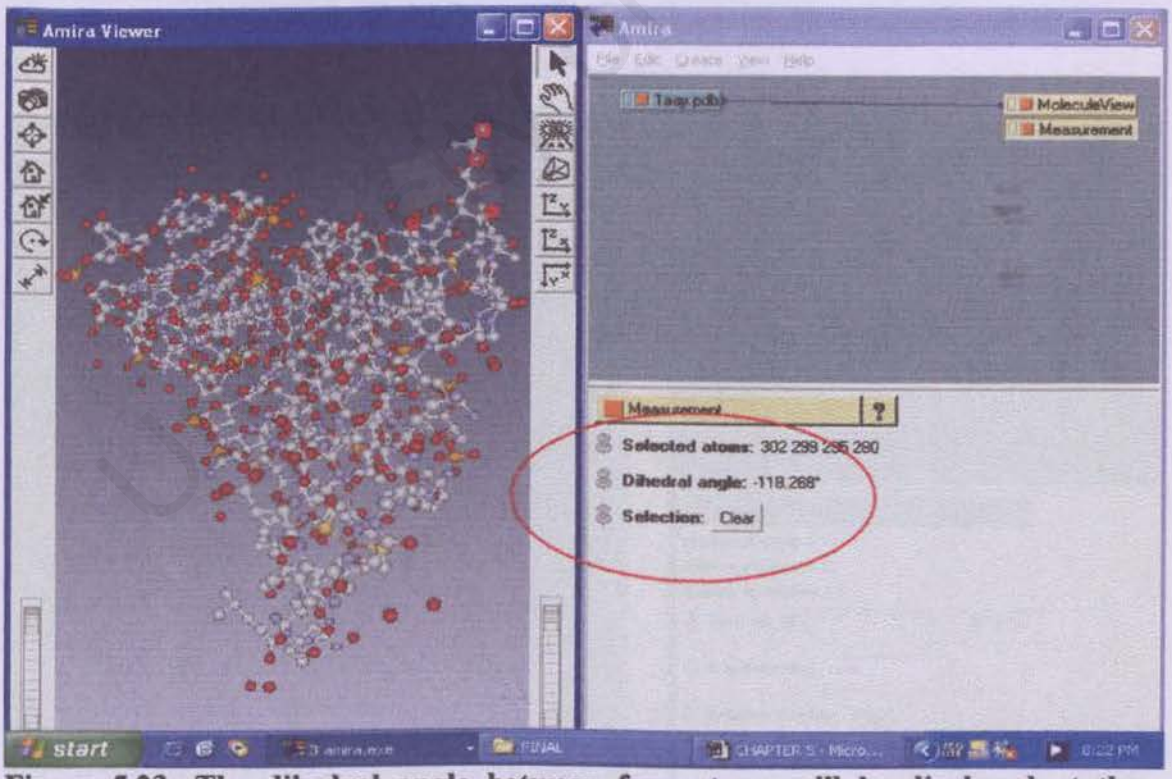


Figure 5.23: The dihedral angle between four atoms will be displayed at the working area of Amira if four atoms are selected at the viewer

2. Comparing two Molecules

Purpose: To view the same molecule in different time-steps. This task applies only to molecules with series of time steps, such as alkane molecules with suffix .zmf.

Molecule data that are used in this task: alkane.zmf

Load the .zmf data into Object Pool-> Select MolTrajectory-> Select Molecule-> Open MoleculeView -> select Mode to Balls and Sticks.

Back to MolTrajectory-> Select Molecule. This will create an entry Molecule2-> Open MoleculeView2-> select Mode to Balls and Sticks. The same molecule will appear at the viewer-> at Time, click the arrow ▶. The time-step will increase to 2. At viewer, noticed that the second molecule will slightly displace. This new position shows the molecule at TimeStep 2.

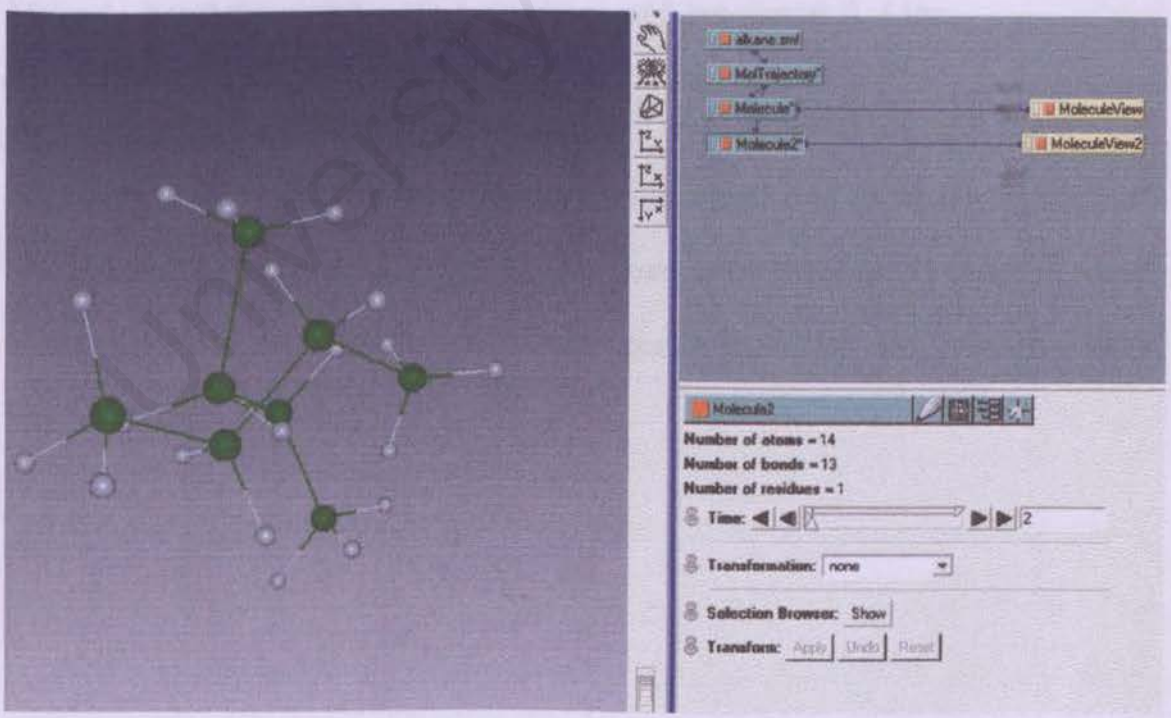


Figure 5.24: The position of the same molecule shown on different time-step

3. Computing Mean Molecule

Purpose: This module computes the *mean* molecule of a molecular trajectory by averaging the atom positions. In order to do this, the molecules need to be transformed to a common coordinate system. This can be done either by aligning all time steps to some reference molecule.

AlignMaster-In order to compute the mean molecule, all molecules of the trajectory need to be aligned with respect to a certain molecule. This is done by right-clicking on the little rectangle of the module icon in the object pool, selecting *AlignMolecule*, and connecting it to a molecule in the object pool.

PrecomputedAlignment-Instead of aligning all time steps of a trajectory to the *AlignMaster*, user can use a precomputed alignment to align the time steps.

Molecule data that are used in this task: alkane.zmf, but_cluster_3_1.idx

Back to MolTrajectory, remove Molecule and Molecule2-> load data file but_cluster_3_1.idx -> compute -> MeanMolecule-> right click on the left rectangle of MeanMolecule icon-> select AlignMaster-> connect blue line to Molecule-> select MeanMolecule-> press All of the Select port-> press Do it-> visualize with MoleculeView.

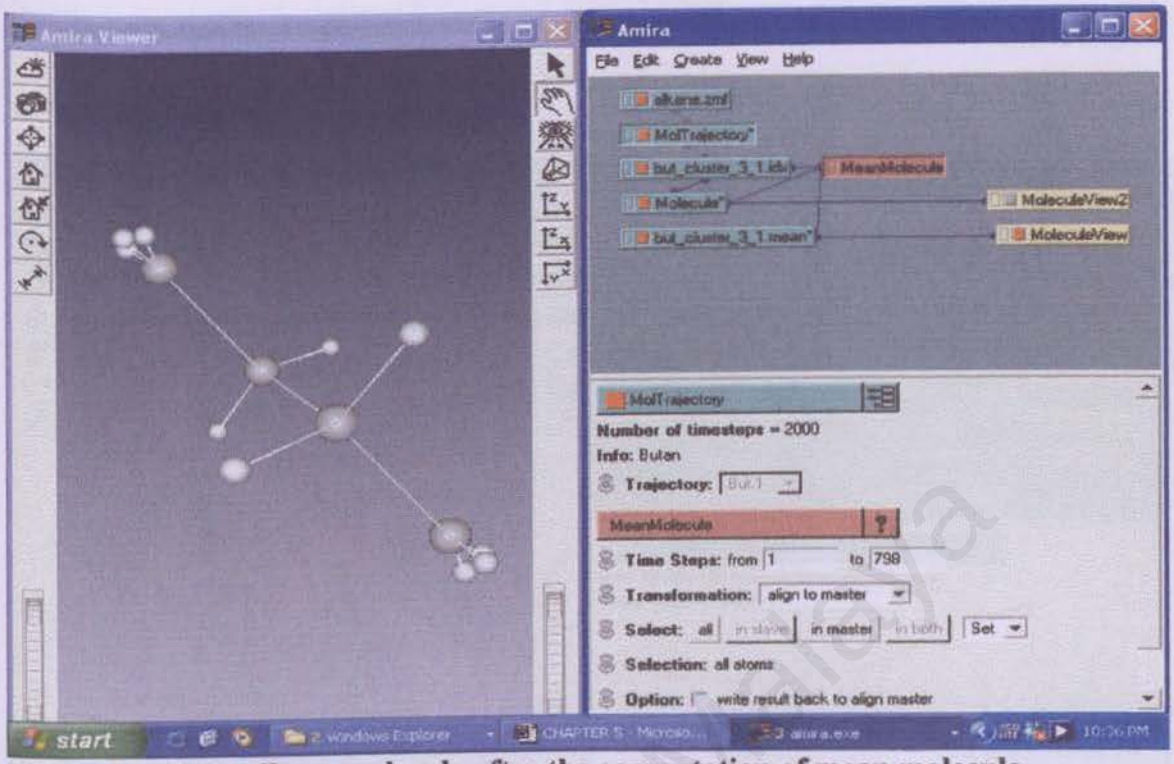


Figure 5.25: The alkane molecule after the computation of mean molecule

4. Computing and Visualizing a Configuration Density

Purpose: The Configuration Density module gives an impression of the fuzziness of the conformation by computing a probability density for the positions of atoms and bonds within a molecular trajectory. This density can be visualized with the Isosurface and Voltex modules.

Molecule data that are used in this task: alkane.zmf, but_cluster_3_1.idx

Remove the MeanMolecule icon-> Back to but_cluster_3_1.idx-> left click-> Compute-> ConfigurationDensity-> connect AlignMaster to but_cluster_3_1.mean-> click on ConfigurationDensity icon, at the working area, press All button from Select port-> press Field button of Compute port to compute Density-> to visualize, select

Display-> Isosurface/Voltex-> set Threshold value for Isosurface module-> press Do It at Action port.

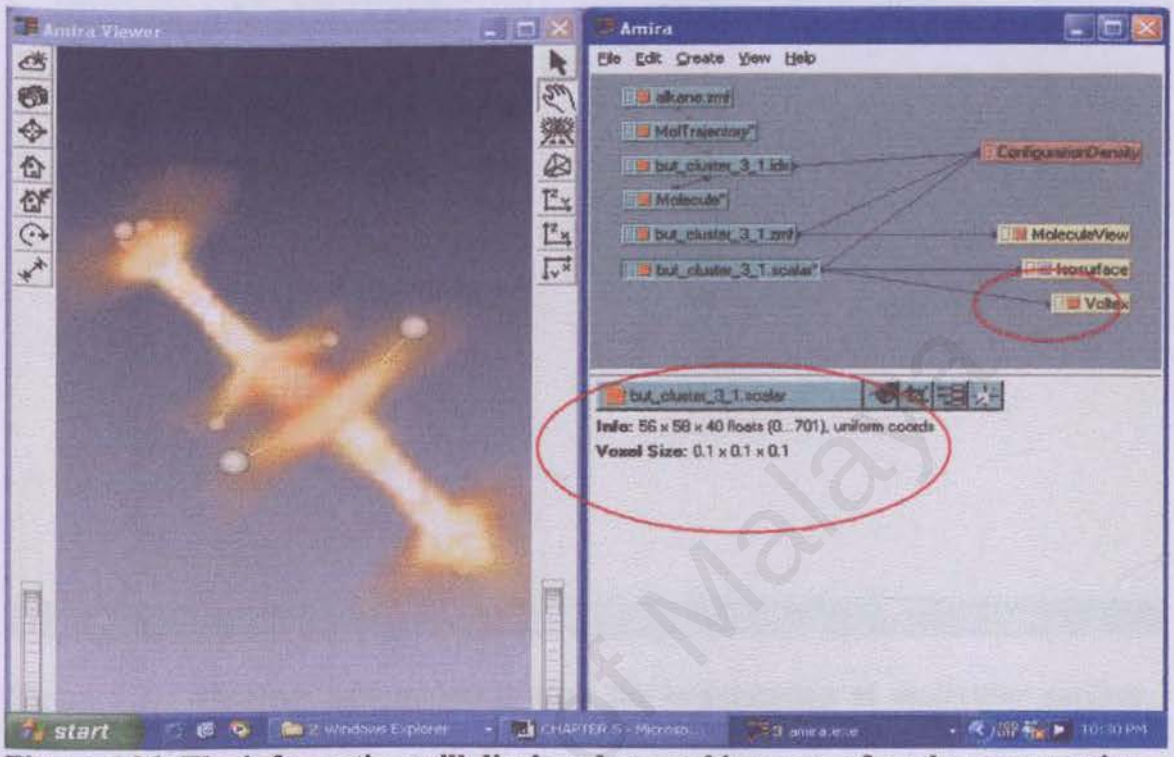


Figure 5.26: The information will displayed at working area after the computation of configuration density. The molecule has been visualized using Voltex module in 2D texture mode

5. Compute Molecular Surface

Purpose: To compute molecular surfaces with different resolutions.

Molecule data that are used in this task: 1aay.pdb

Load data-> right click on green icon-> Compute-> CompMolSurface-> click on CompMolSurface icon-> Press Do It at Action port-> Attach MolSurfaceView to the newly created green icon, XXX-surf-> for a better resolution, increase the number of points per A^2-> press Do It.

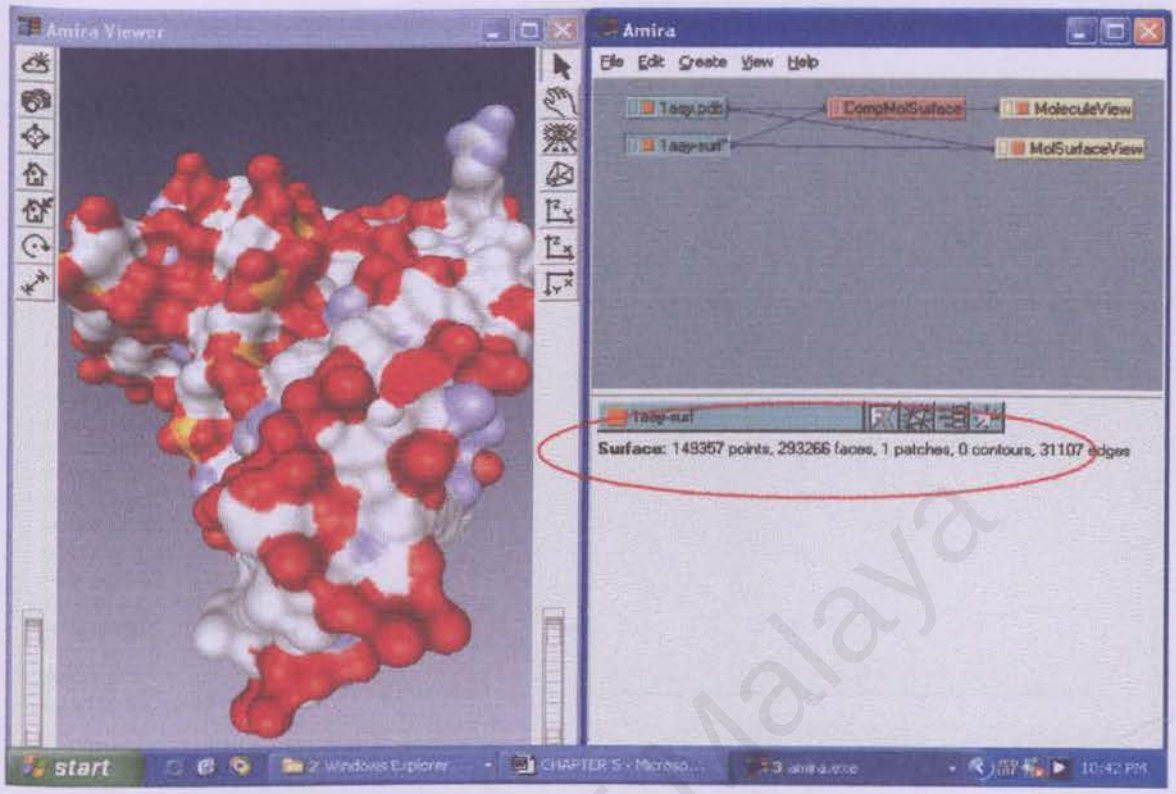


Figure 5.27: The 1aay.pdb molecule after the computation of molecular surface and visualized using molecule surface view with resolution value 20 and set according to atom number color. The details of surface computation are shown on the working area

6. Compute Partial Surface

Purpose: To compute surface contribution for selected atoms.

Molecule data that are used in this task: 1aay.pdb

Open selection browser-> type within (residues/XXX) in the browser's Expression command line-> press Replace button-> At CompMolSurface-> Quality port-> select correct-> Option port-> select partial surface-> press Do It.

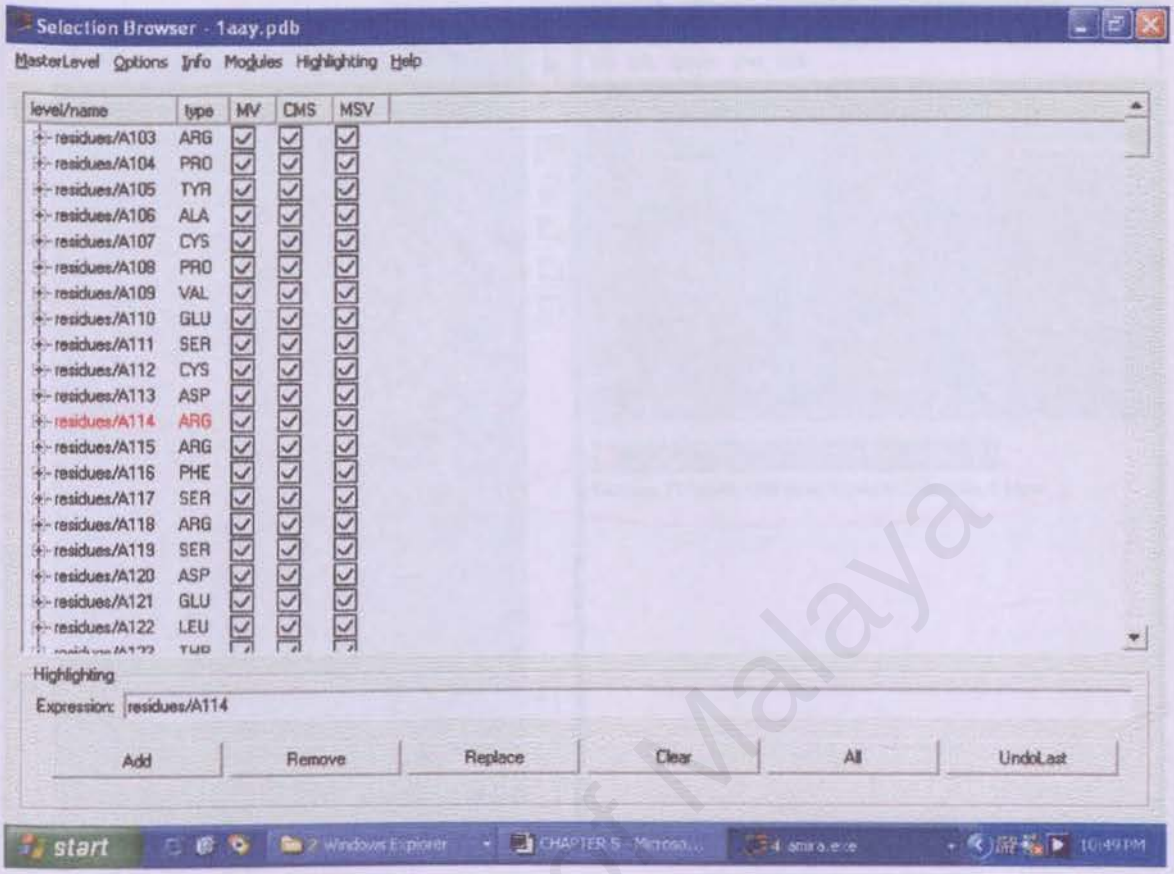


Figure 5.28: The atoms at the residues/A114 have been selected to calculate its surface contribution

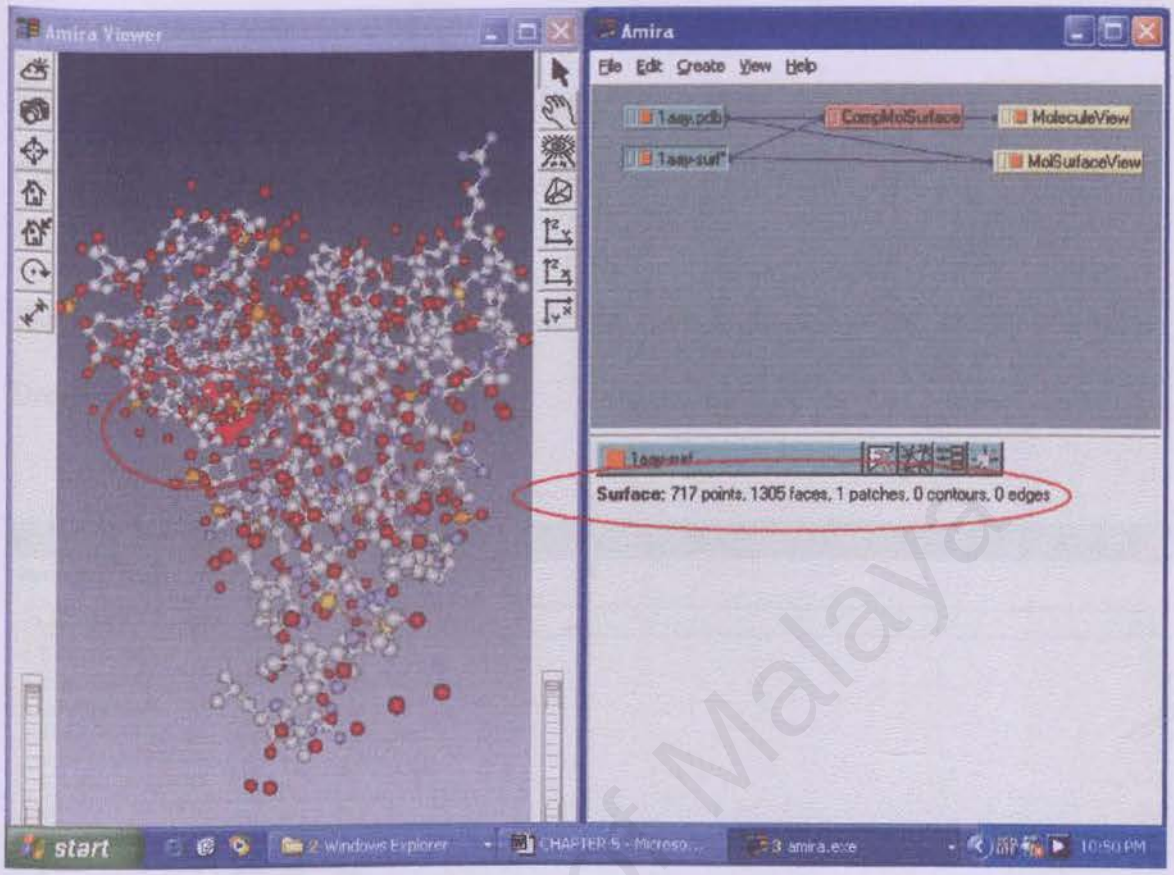


Figure 5.29: The surface contributions of atoms in residues/A114 that have been calculated are shown in the working area. At the viewer, the selected residue has been highlighted automatically

7. Molecular Surface of a Restricted Sets of Atoms

Purpose: In most cases, some molecules contain water molecules, which in most cases not desired when computing the molecular surface. Here, it eliminates the water molecules.

Molecule data that are used in this task: 1aay.pdb

Go to selection browser-> on Type column, click HOH-> all HOH will be selected-> at browser, right click on heading CMS(CompMolSurface)-> select Remove-> at CompMolSurface-> deselect Partial surface-> press Do It button.

For the new surface, only those residues that have a check marked in the Selection Browser were considered.

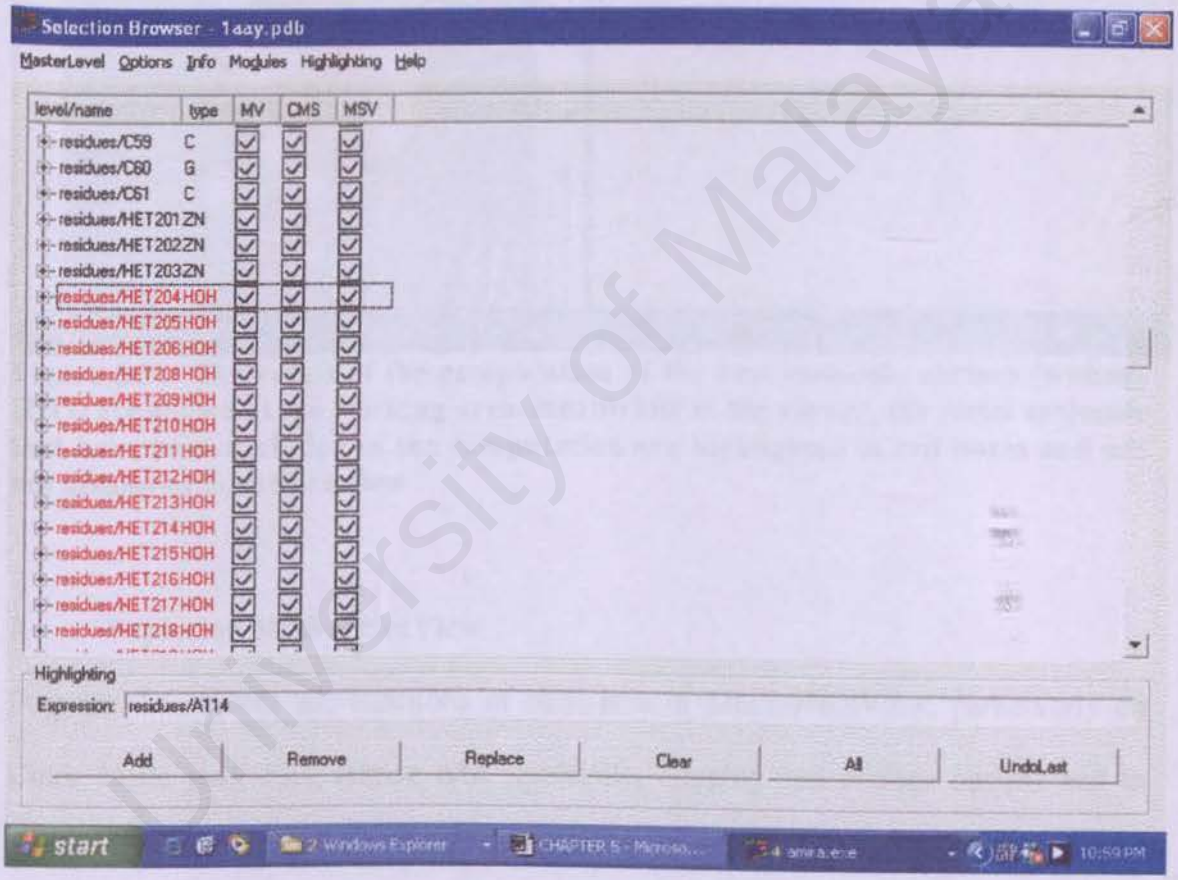


Figure 5.30: The water molecules, H2O have been selected (highlighted in red) at the selection browser to eliminate them from the computation of new surface contribution

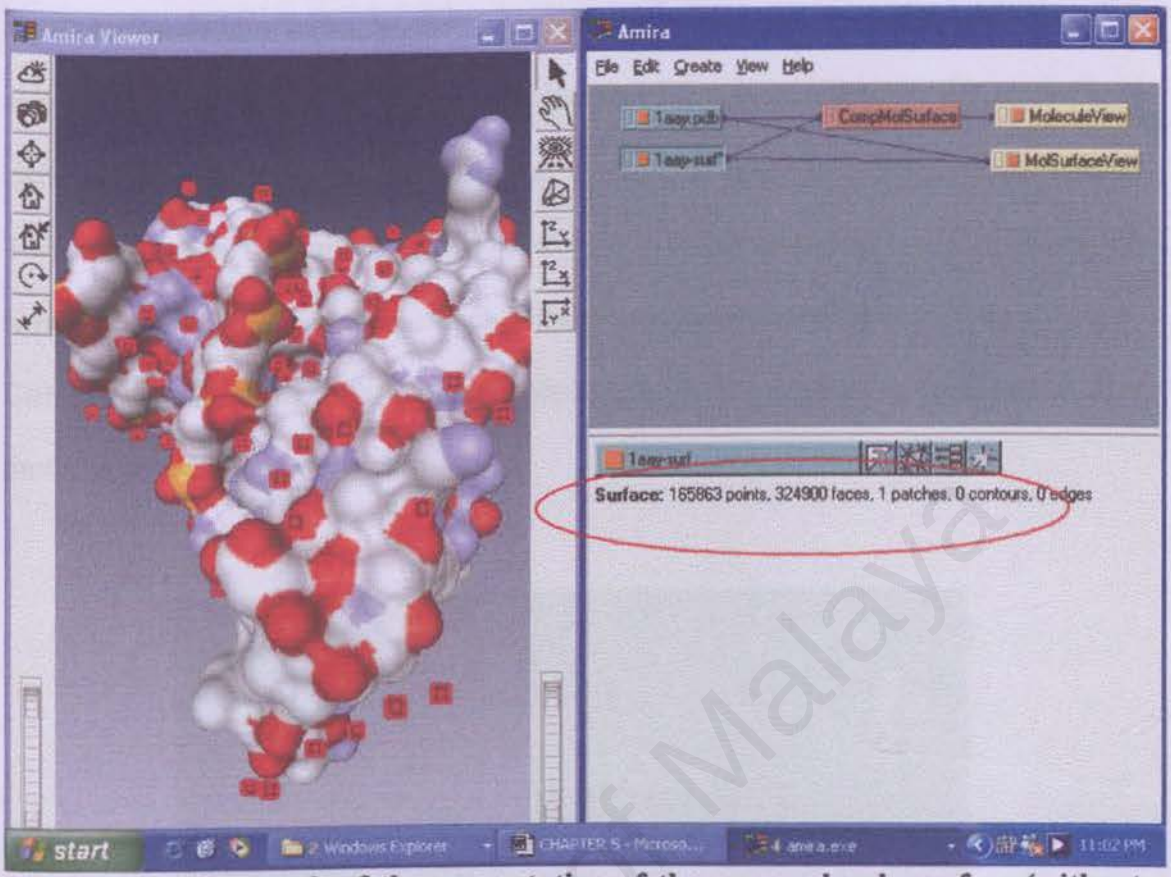


Figure 5.31: The result of the computation of the new molecule surface (without H2O) are shown at the working area, meanwhile in the viewer, the water molecule that have been excluded in the computation are highlighted in red boxes and are not displayed in surface view

8. Exploring MolSurfaceView

Purpose: To explore the functions of each port in MolSurfaceView, particularly on Color Mode port, Pick Action port (molecule, clipping and surface option) and to familiarize with Highlighting and Buffer Port.

Molecule data that are used in this task: 1aay.pdb

Compute the whole molecular surface with a resolution of 2 points per-> click on the MolSurfaceView icon to see its user interface in the working area-> select molecule for

the Color Mode port-> change the Color port's first menu entry to residues-> select an appropriate color map for the Discrete CM port by right-clicking on the color bar-> switch to interaction mode-> clicking on the surface with the middle mouse button. This will displays information about the atom that user has clicked on in the upper left corner of the viewing window as long as the user keep the mouse button pressed. If user Ctrl-click, the information will remain displayed even after releasing the mouse button until the next mouse click on the surface.

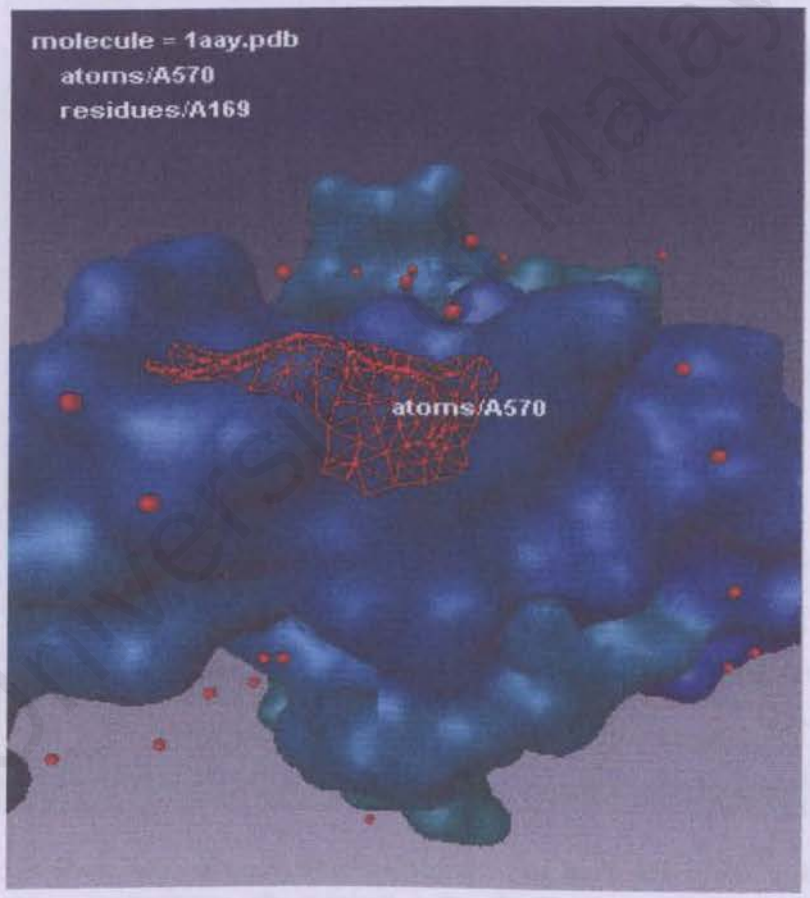


Figure 5.32: DNA molecule that has been visualized in surface view with resolution value 2. The selected atom, residues in which it belongs and the molecule's details are displayed when user Ctrl-click on the molecule

(continue from previous)
-> at Highlighting port, press Clear button to remove the selection-> change the Pick Action port to clipping-> pick any triangle of the surface.

All triangles further away from the picked triangle than the distance given by the Selection Distance port will be cut off. All triangles within this distance will remain, however, only if they are connected to the picked triangle without leaving the sphere around the picked point.

Next, press the All button of the Buffer port to display the whole surface-> change the Pick Action port to surface-> shift-click on the surface in the viewing window.

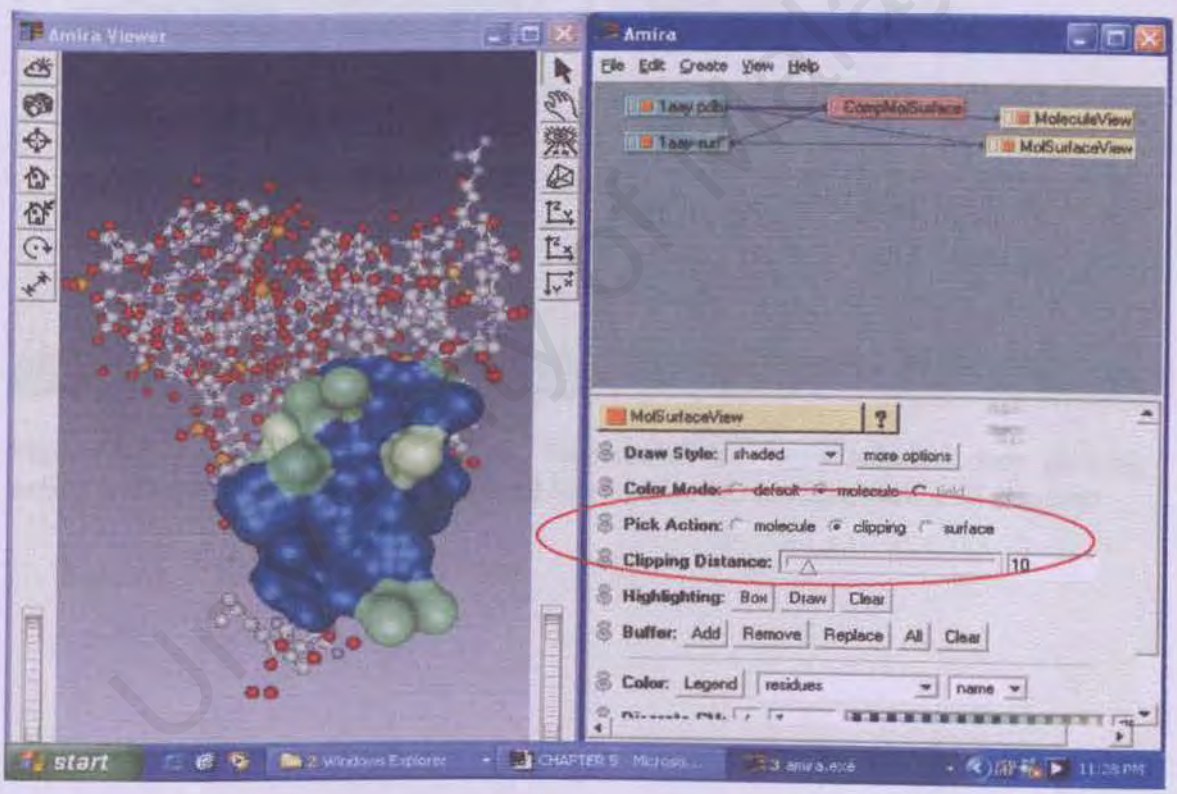


Figure 5.33: In Pick Action port, the "clipping" has been selected. Noticed that all triangles further away from the picked triangle than the distance given by the Selection Distance port will be cut off. All triangles within this distance will remain, however, only if they are connected to the picked triangle without leaving the sphere around the picked point

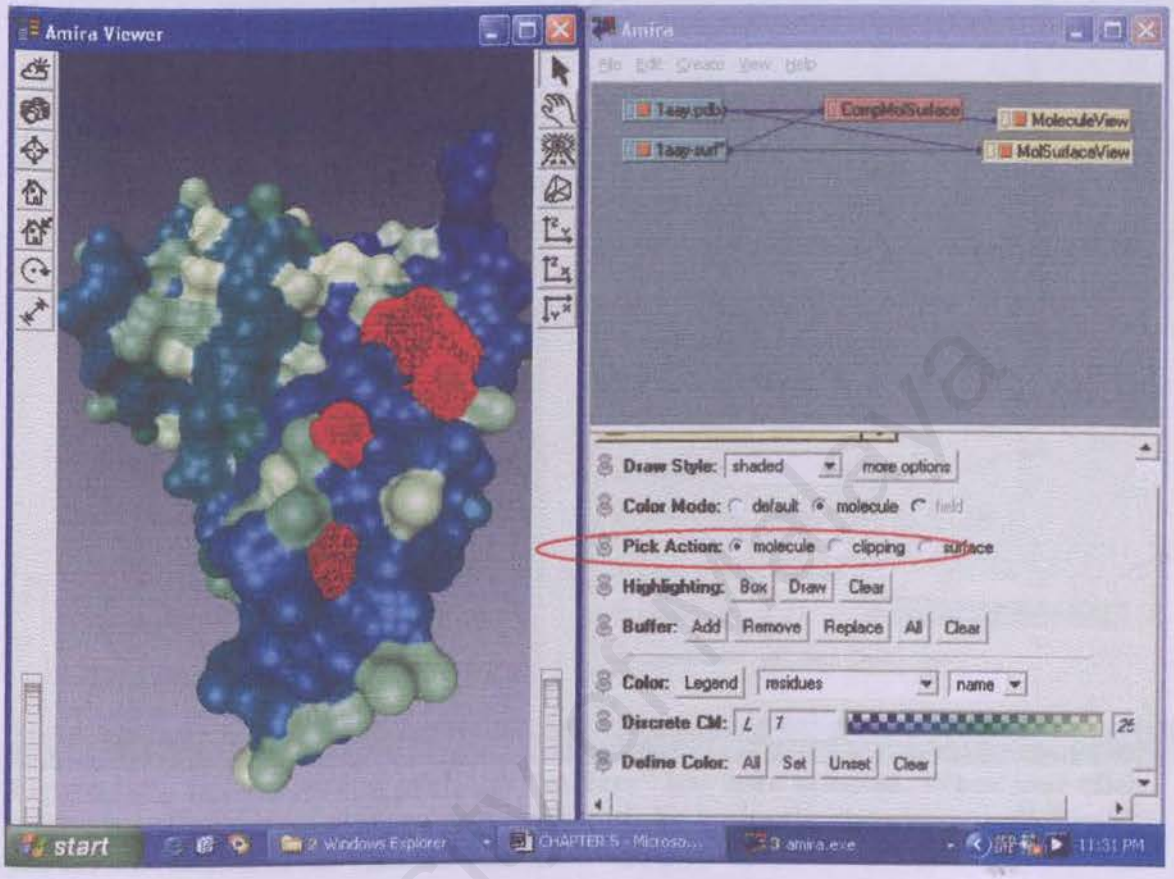


Figure 5.34: In Pick Action port, the "molecule" has been selected. Any picking action will highlight the residues in red lines while the surface view still remains

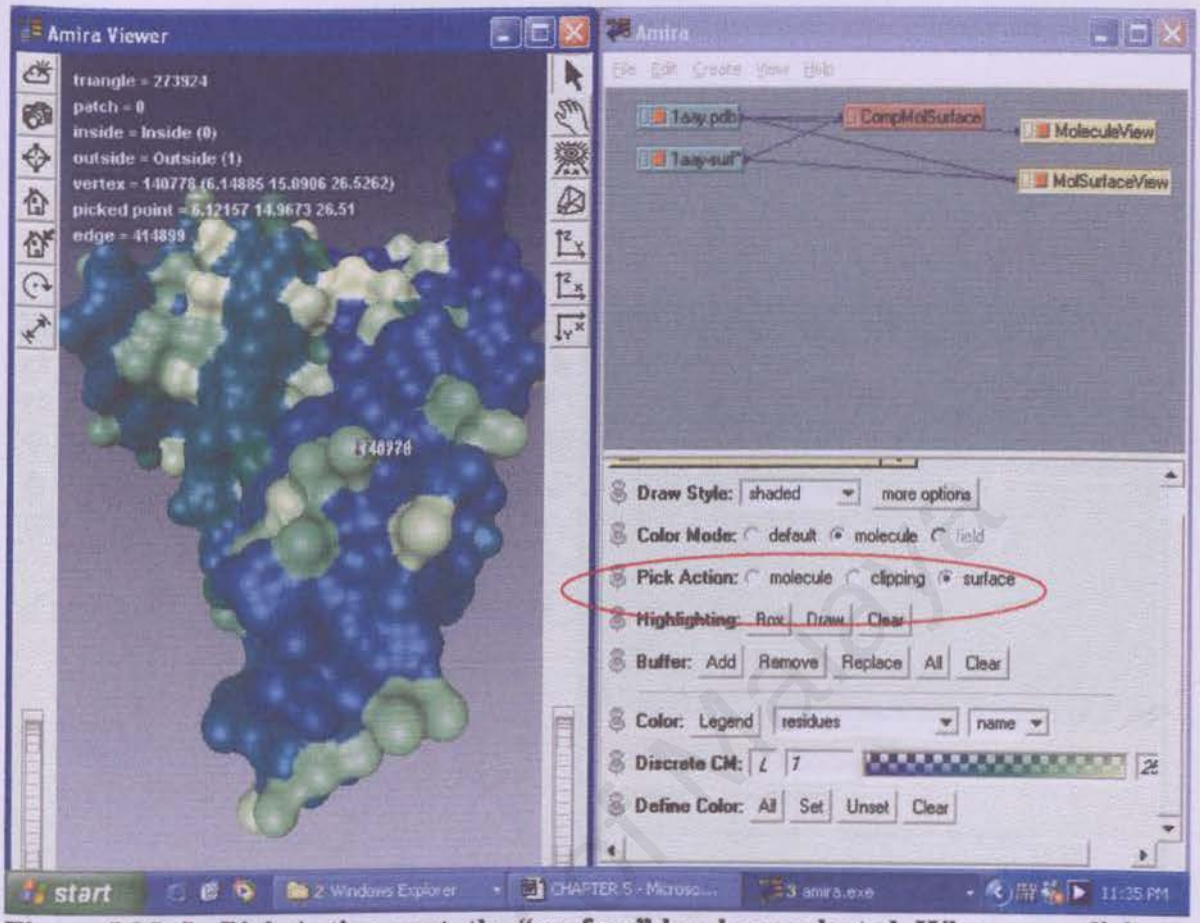


Figure 5.35: In Pick Action port, the "surface" has been selected. When user click using the middle mouse button at any part of the molecule, the viewer will display the surface details of the selected part of the molecule

9. Sequence Alignment

Purpose: To align sequences of two molecules. It facilitates the comparison of two sequences. This task applied to both proteins and nucleic acid except for t-rna molecules containing modified bases.

Molecule data that are used in this task: 1IGM.pdb, 2JEL.pdb

Right click on XXX.pdb icon-> select Compute-> select AlignSequences-> right click on the rectangle at the AlignSequences icon-> a pop up menu will appear displaying all connection ports opens-> select MoleculeB-> connect the port to another XXX.pdb file

icon by clicking on it-> on AlignSequence icon, Align Type port, choose semiglobal from the second menu-> press AlignSequences button-> press AlignSequence port.

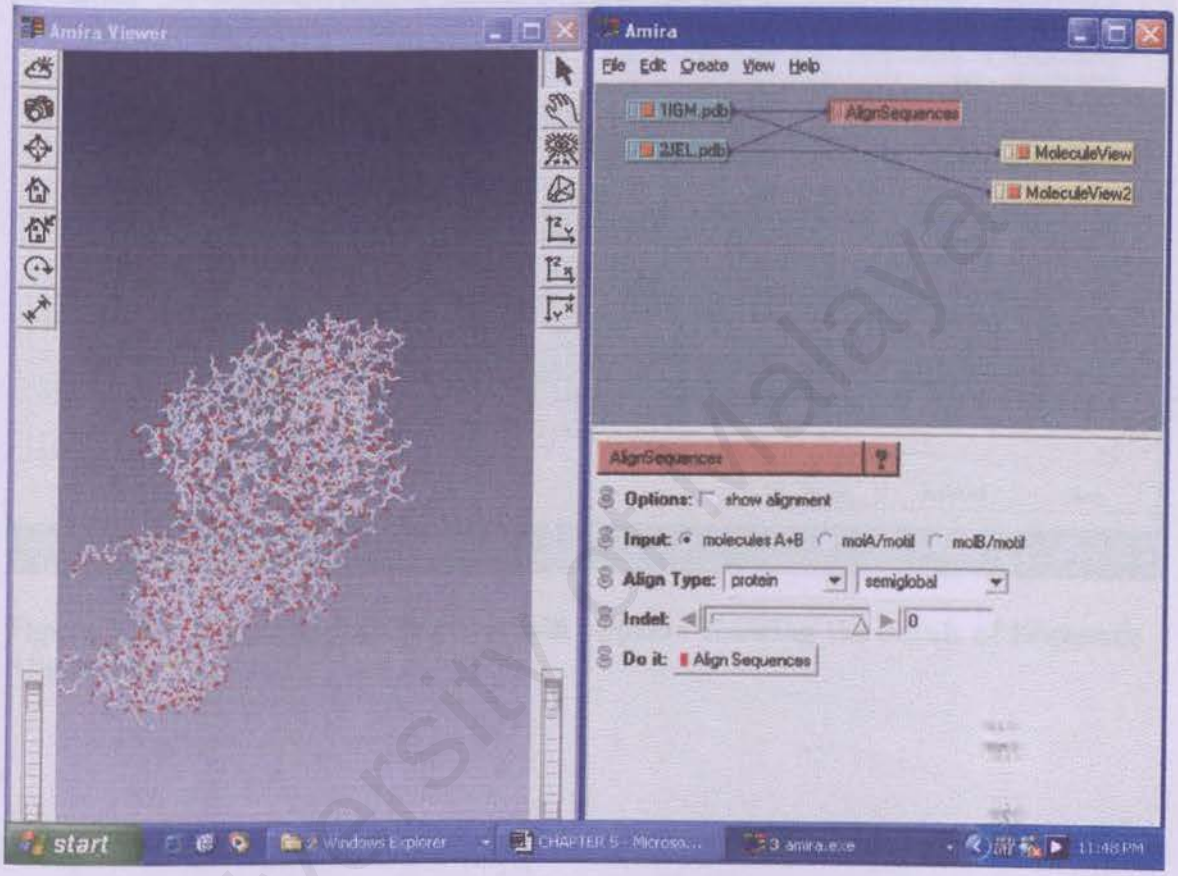


Figure 5.36: In sequence alignment, the first molecule (1IGM.pdb) is aligned with second molecule (2JEL.pdb) through Align Master. After aligned, a blue line will be connected between Align Sequence icon and the second molecule

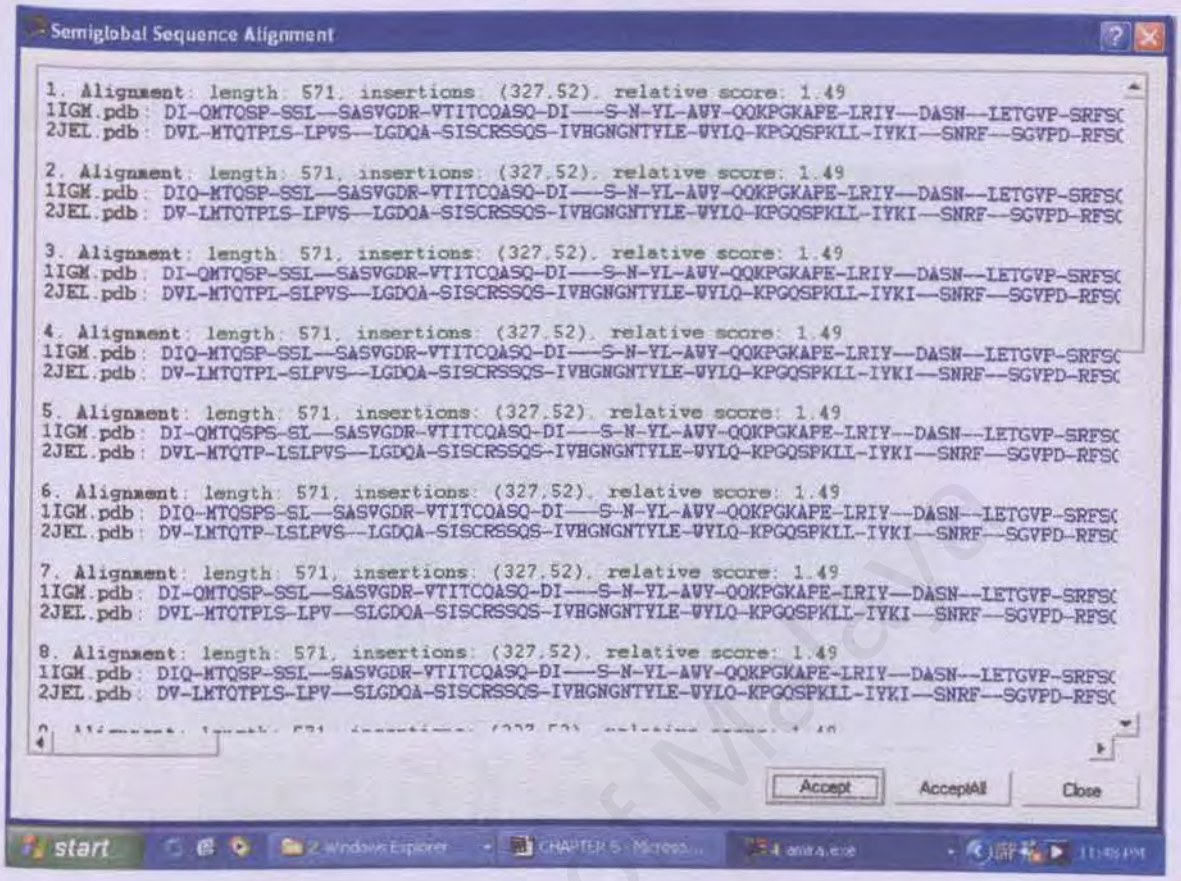


Figure 5.37: A new pop-up window will appears showing the result of Sequence Alignment

10. Editing of Molecules

Purpose: To use Molecule Editor in editing molecules. It allows adding new bonds or changing the overall topology of the molecule as well as manipulating Cartesian or internal coordinates.

Molecule data that are used in this task: 1HVRm.pdb

Load data-> To invoke Molecule Editor, click on the green icon-> At working pool click on  -> a window will appear as shown below.

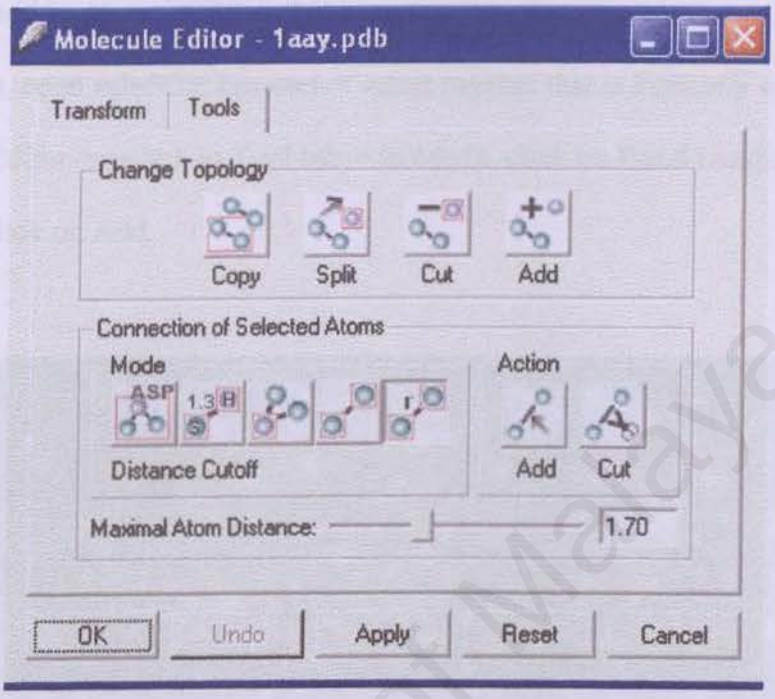


Figure 5.38: The pop-up window (Tools tab)

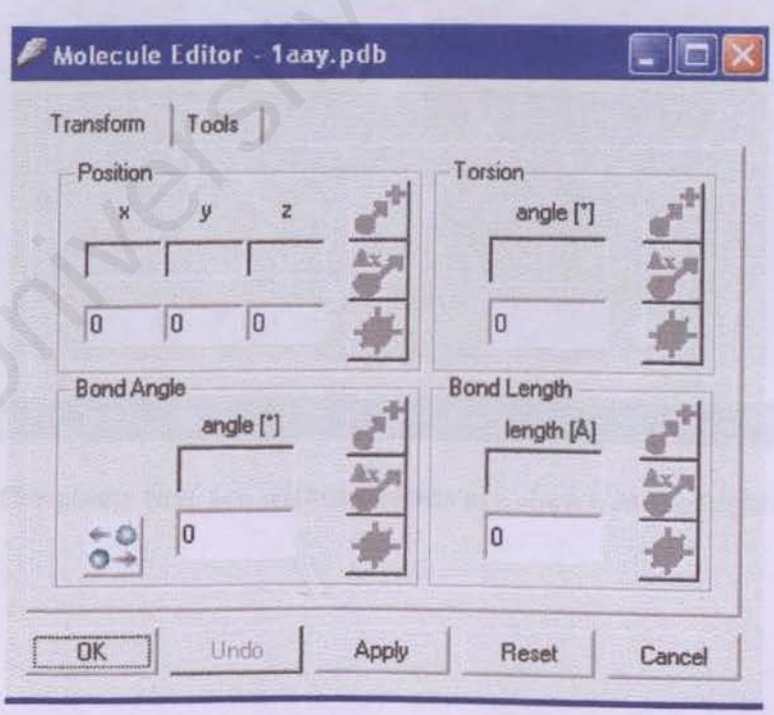


Figure 5.39: The pop-up window (Transform tab)

Among the task that can be performed in molecule editor are:

i. Adding bonds to a part of molecule

Purpose: To add bonds to inhibitor that is currently without bonds.

To add bonds, open selection browser-> select residue that is currently without bond-> at Molecule Editor-> switch to Tool tab-> In Mode, click on Bond Length table button-> In Action click on Add.

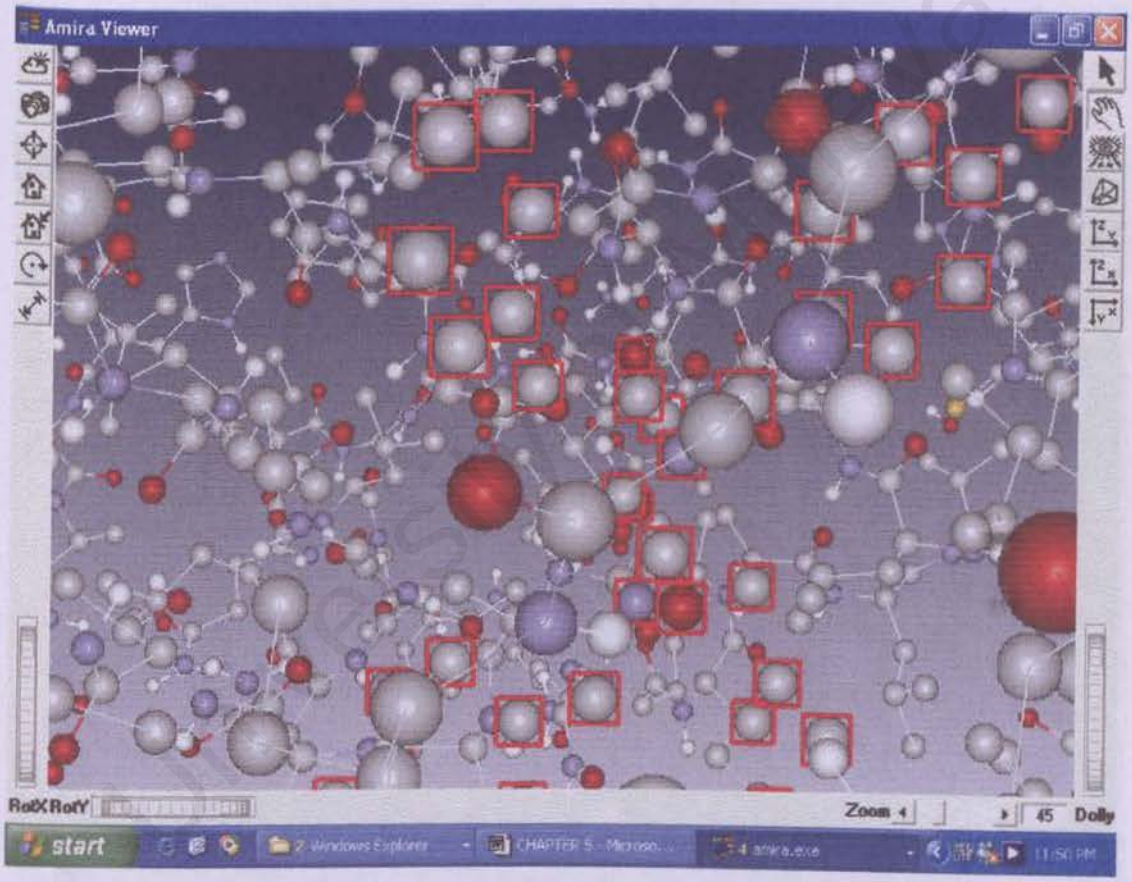


Figure 5.40: The atoms that are without bonds are shown in the highlighted box

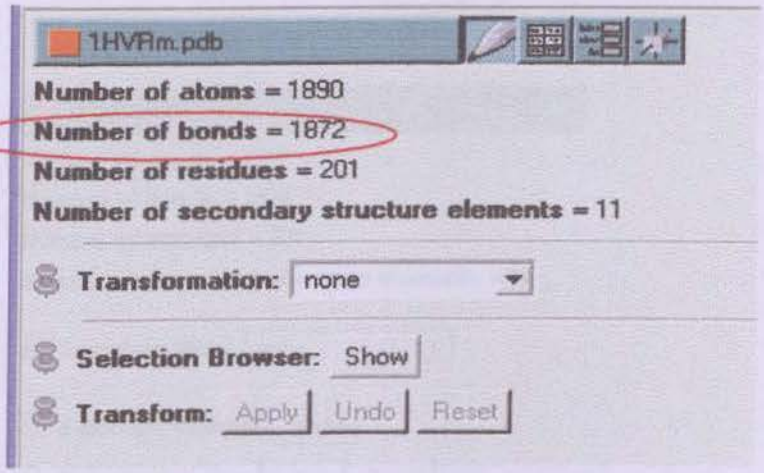


Figure 5.41: The number of bonds (before adding bonds) is shown clearly at the working area

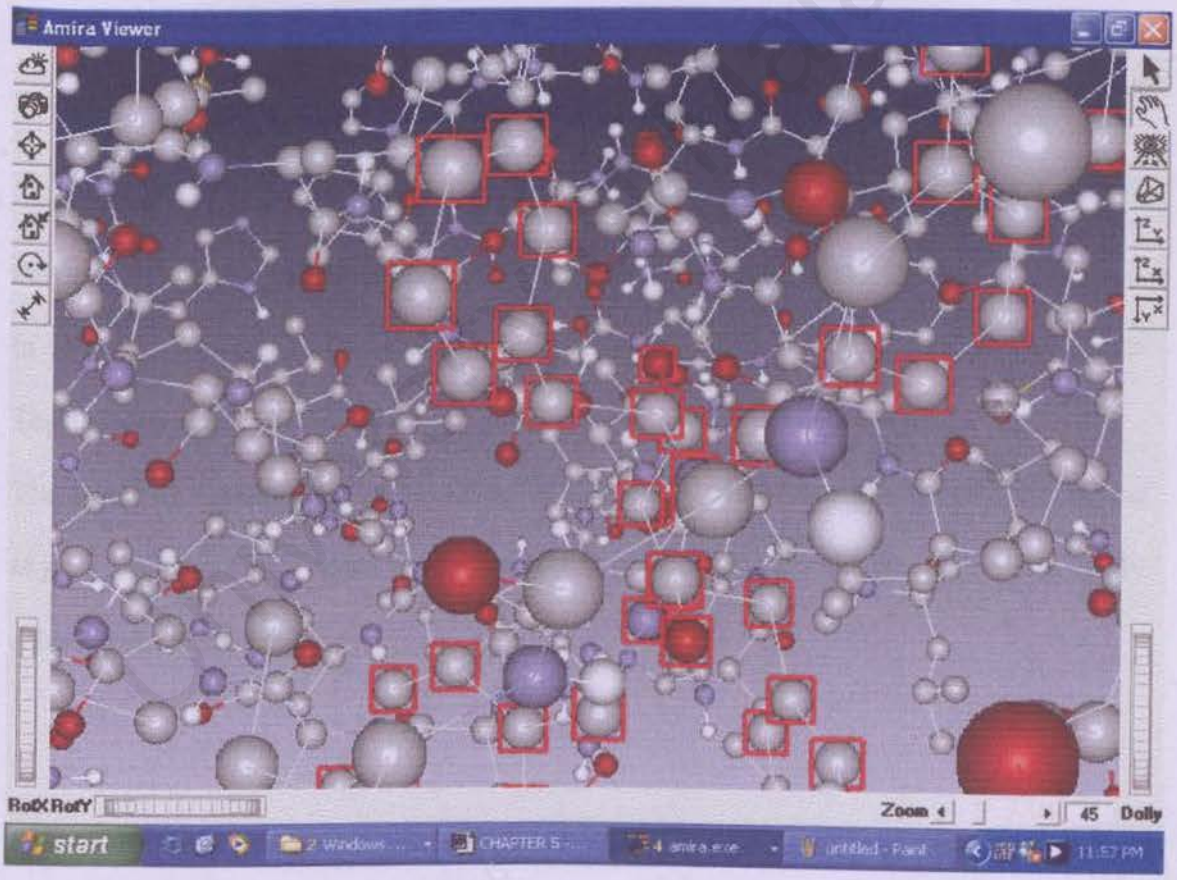


Figure 5.42: The atoms after adding bonds. Noticed the sticks (bonds) are connected to each balls (atoms) that are highlighted earlier

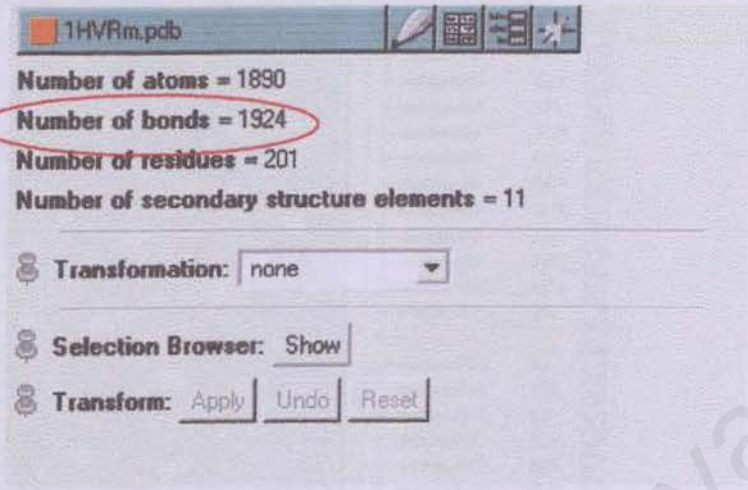


Figure 5.43: The number of bonds has increased after adding bonds

ii. Splitting the molecule

Purpose: To split a molecule into inhibitor and protein (called receptors).

In selection browser, select the residue XXX-> in Molecule Editor, go to Tool tab-> Topology-> Split.

The atoms of the inhibitor are no longer displayed by the MoleculeView and that a new object, XXX2.pdb has appeared in the object pool. This object contains the previously selected atoms of the ligand.

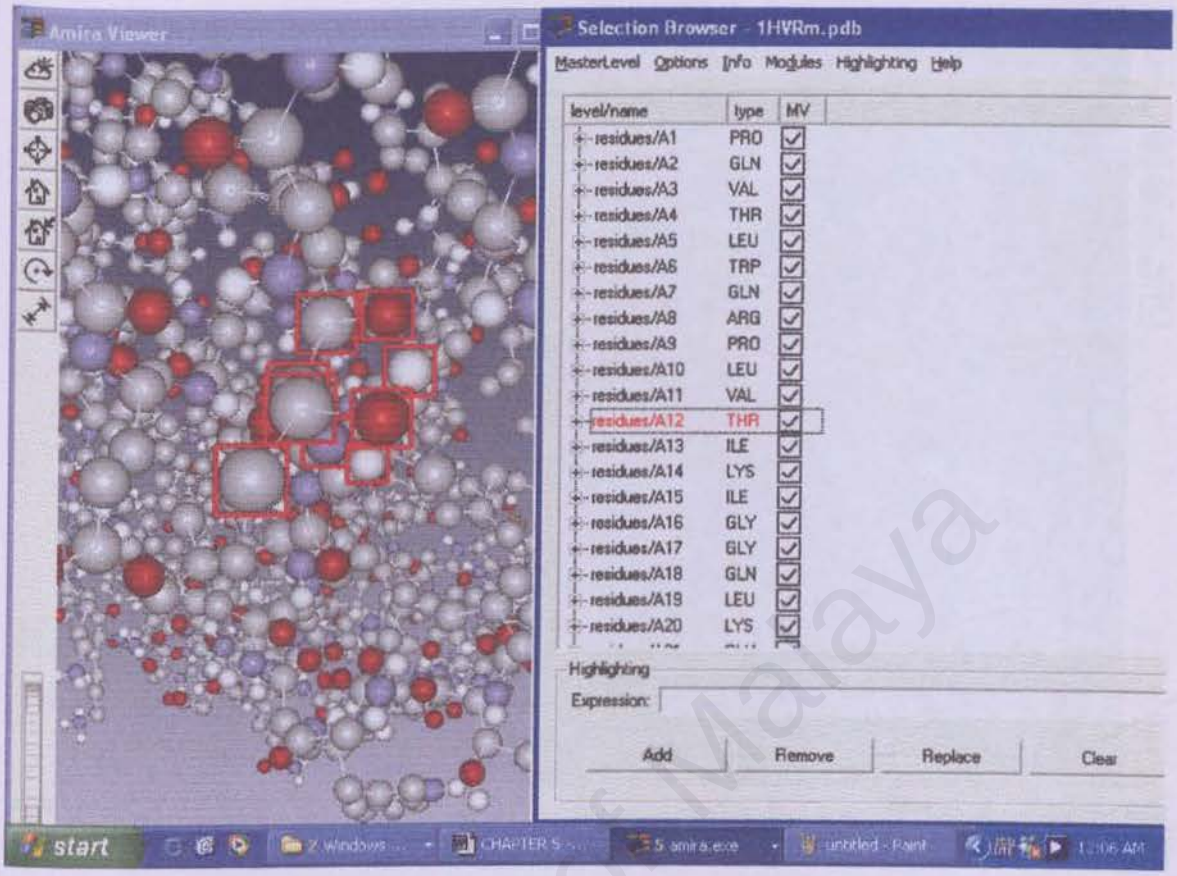


Figure 5.44: The part of molecules that are to be splitted. In the selection browser the residues/A12 are highlighted as in the viewer

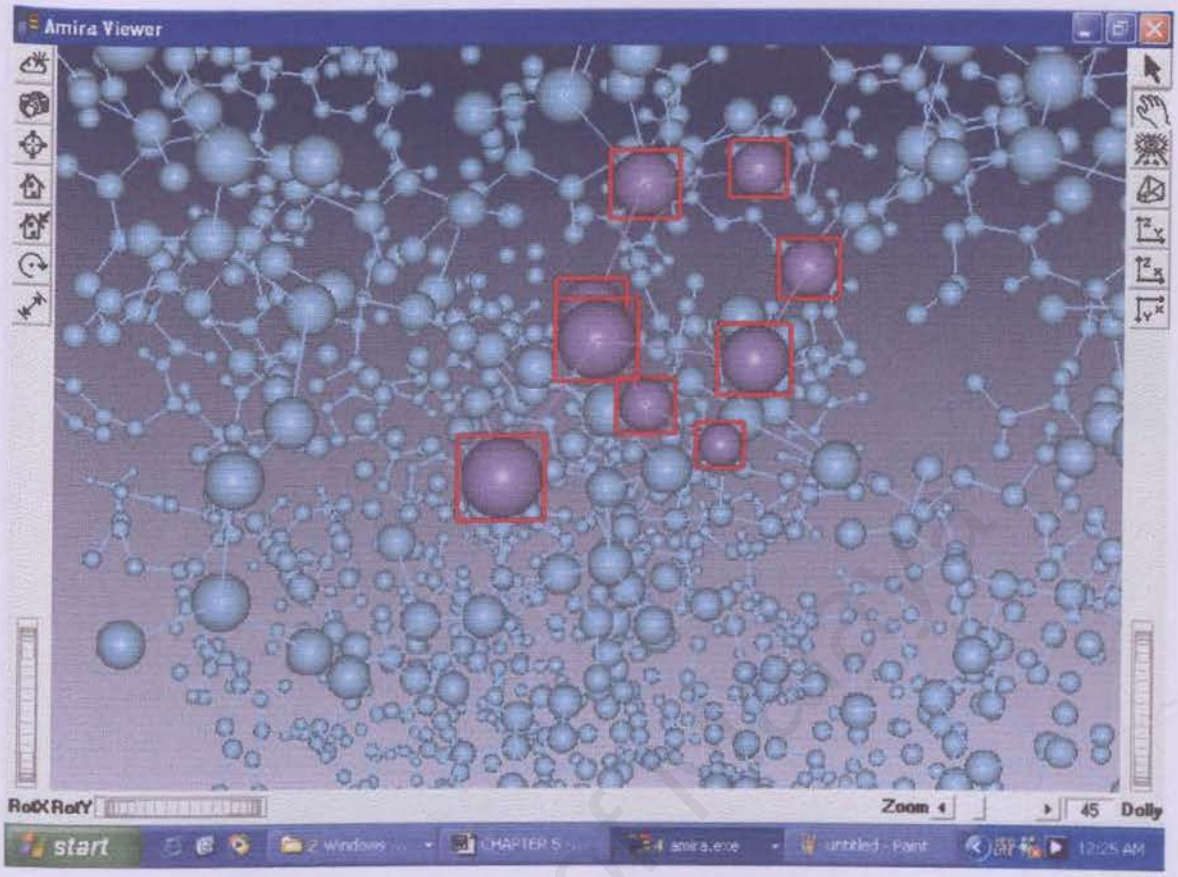


Figure 5.45: To distinguish the atoms that are to be splitted from the molecule, they can be colored according to user's preferences

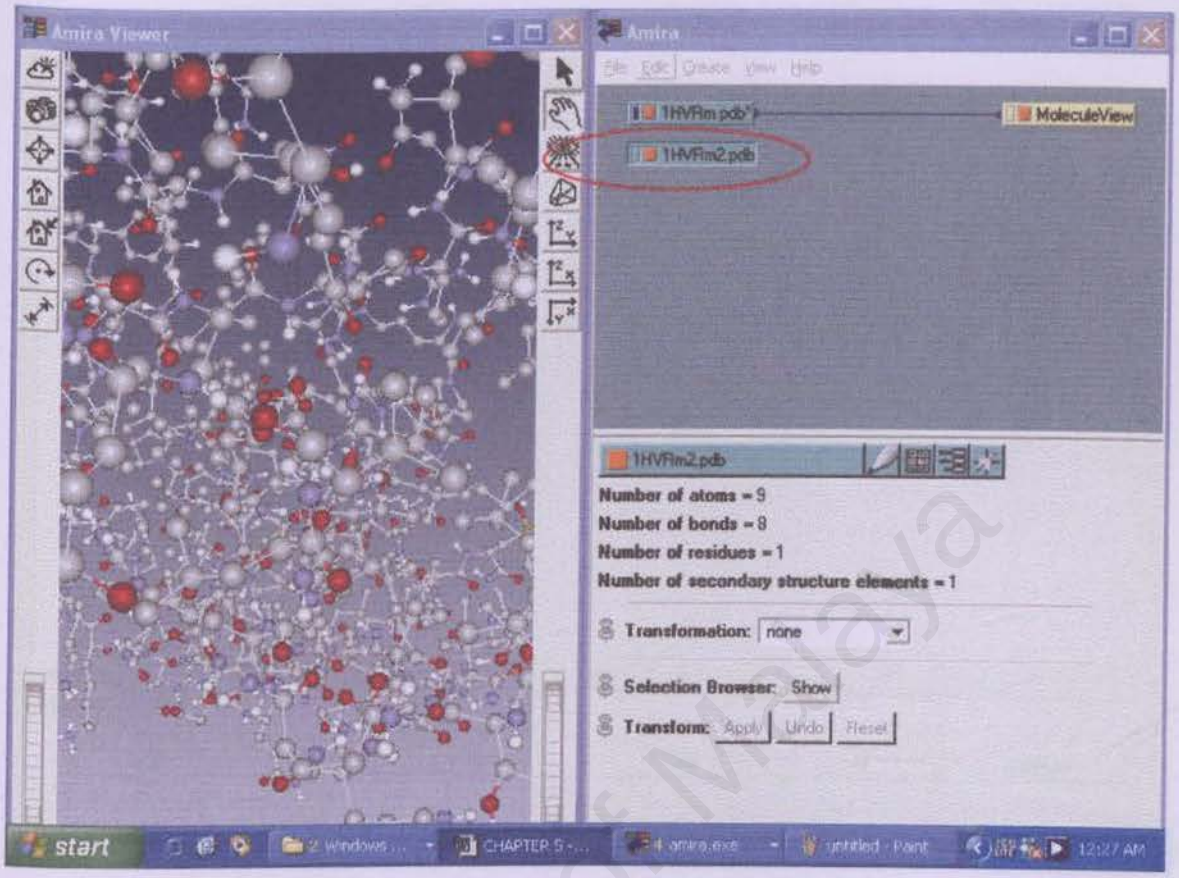


Figure 5.46: The molecule after splitting process. A new icon has appeared at the Object Pool (1HVRm2.pdb)

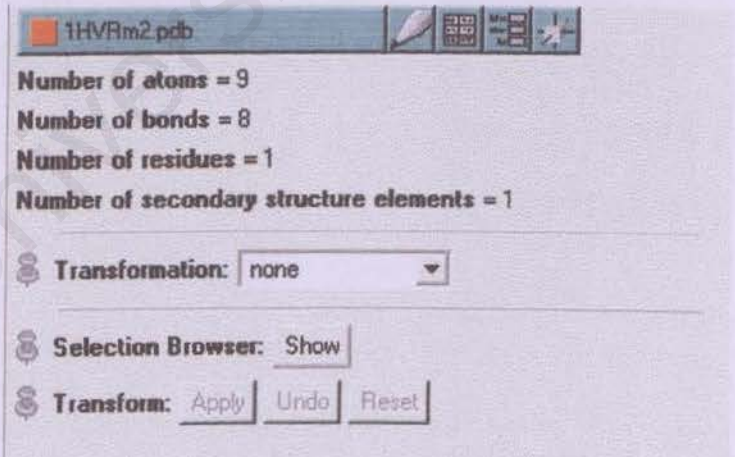


Figure 5.47: At working area, the details regarding 1HVRm2.pdb are shown. This icon represents the residues/A12 that has been splitted from the molecule

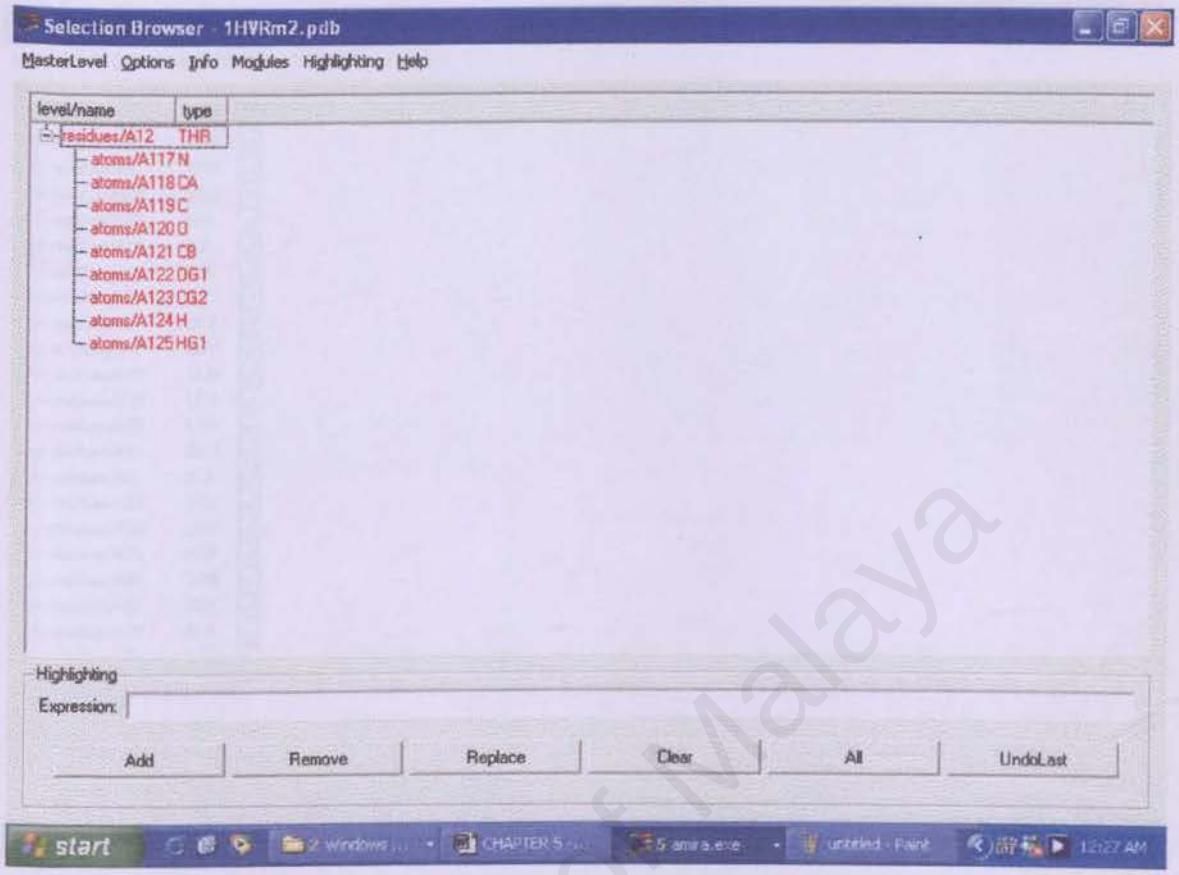


Figure 5.48: The selection browser of 1HVRm2.pdb shows the list of atoms under residues/A12 that have been taken out from the molecule

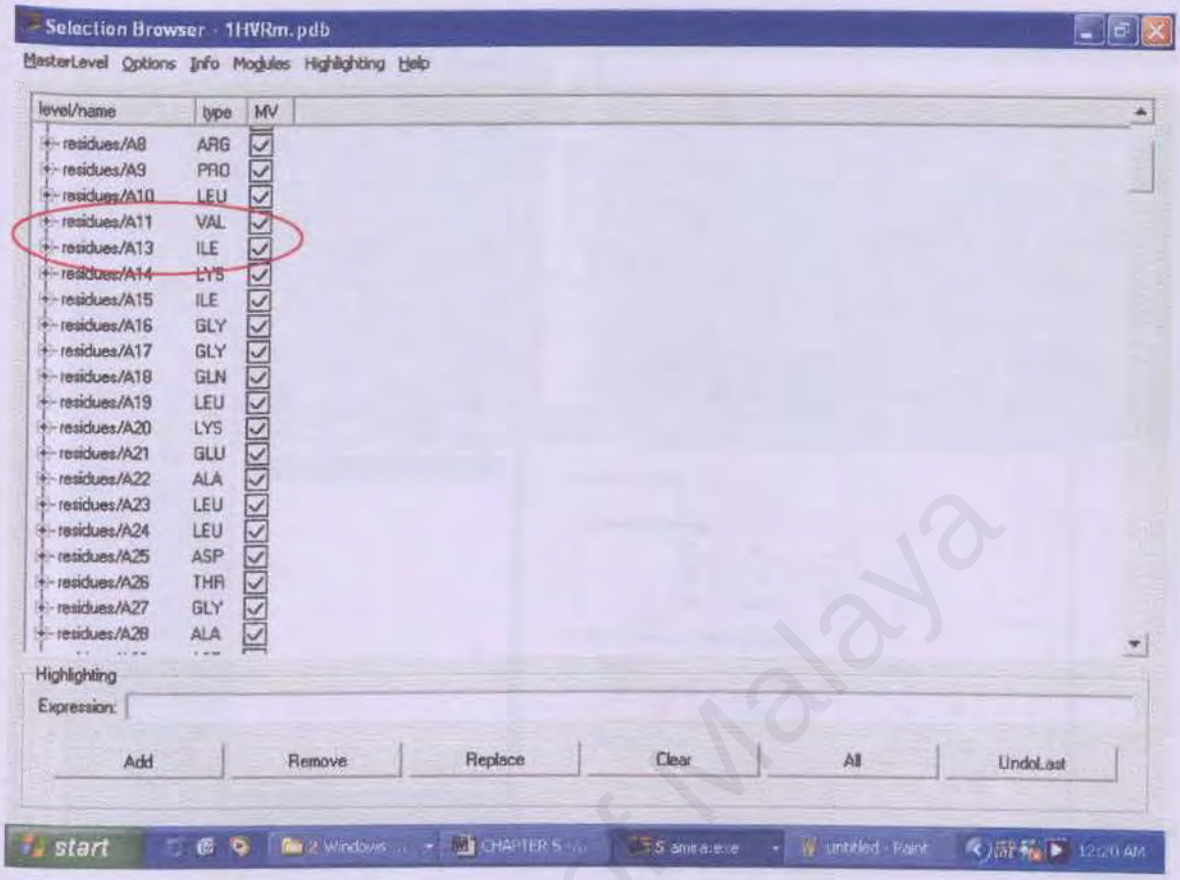


Figure 5.49: The selection browser of 1HVRm.pdb shows the list of residues that are still remained after the splitting process. Noticed that the residues/A12 has been eliminated from the list

iii. Adding another molecule

Purpose: In this case, 1HVRm.pdb entry is a protease (enzyme) which uses up one water molecule to split a polypeptide. Here, water molecule is added to the active site of the enzyme.

Load another molecule, for example, H2O.pdb-> Go to Molecule Editor-> press Add button-> a window will be opened-> all other molecules in the object pool will be shown. Select H2O.pdb molecule and press OK. The atoms of the water molecule have been copied to the XXX.pdb object.

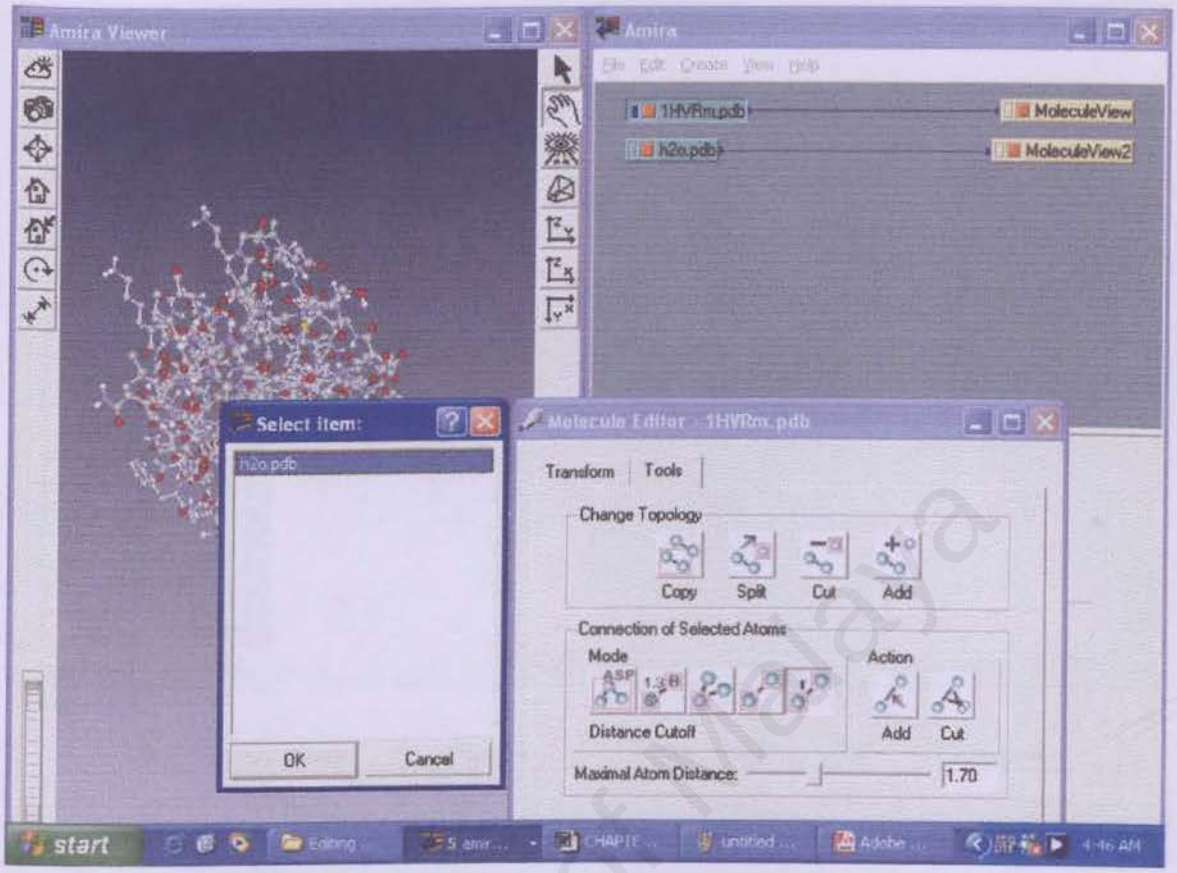


Figure 5.50: Load 1HVRm.pdb and H2O molecule into Object Pool, meanwhile at Molecule Editor, click on Add, a window will pop up showing other molecules in the Object Pool (h2o.pdb)

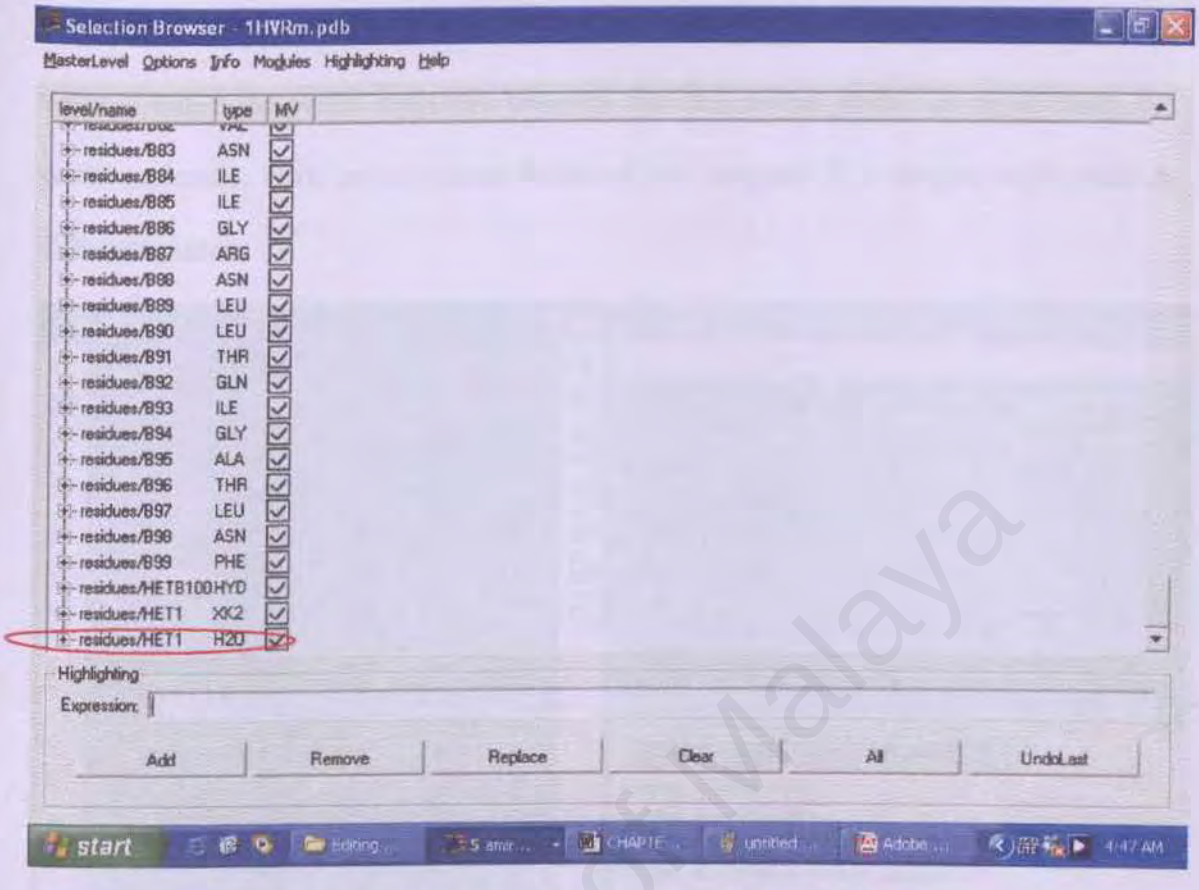


Figure 5.51: The selection browser of molecule 1HVRm.pdb after adding water molecule. Noticed that H2O has been listed into the list of residues in that molecule

iv. Moving parts of the molecule

Purpose: To place the water molecule at the active site of the protease enzyme. To concentrate on the view of region of interest, the molecular view will be reduced to amino acids A25 and B25 between which the water molecule should be placed.

In selection browser, type r/name=?25 OR r/type=H2O-> press Add-> At MV heading, press right button-> activate the Replace option. The viewer will only display the molecules A25 and B25 -> Select water at the selection browser-> at Molecule Editor, go to Transform tab-> show the position of the transform dragger by pressing the

button in the Position section.-> left click on the dragger, and hold on the mouse button down-> move the water molecule between the two amino acids. To re-orientate the water molecule, click on the green knobs of the dragger. The dragger will rotate in different planes.

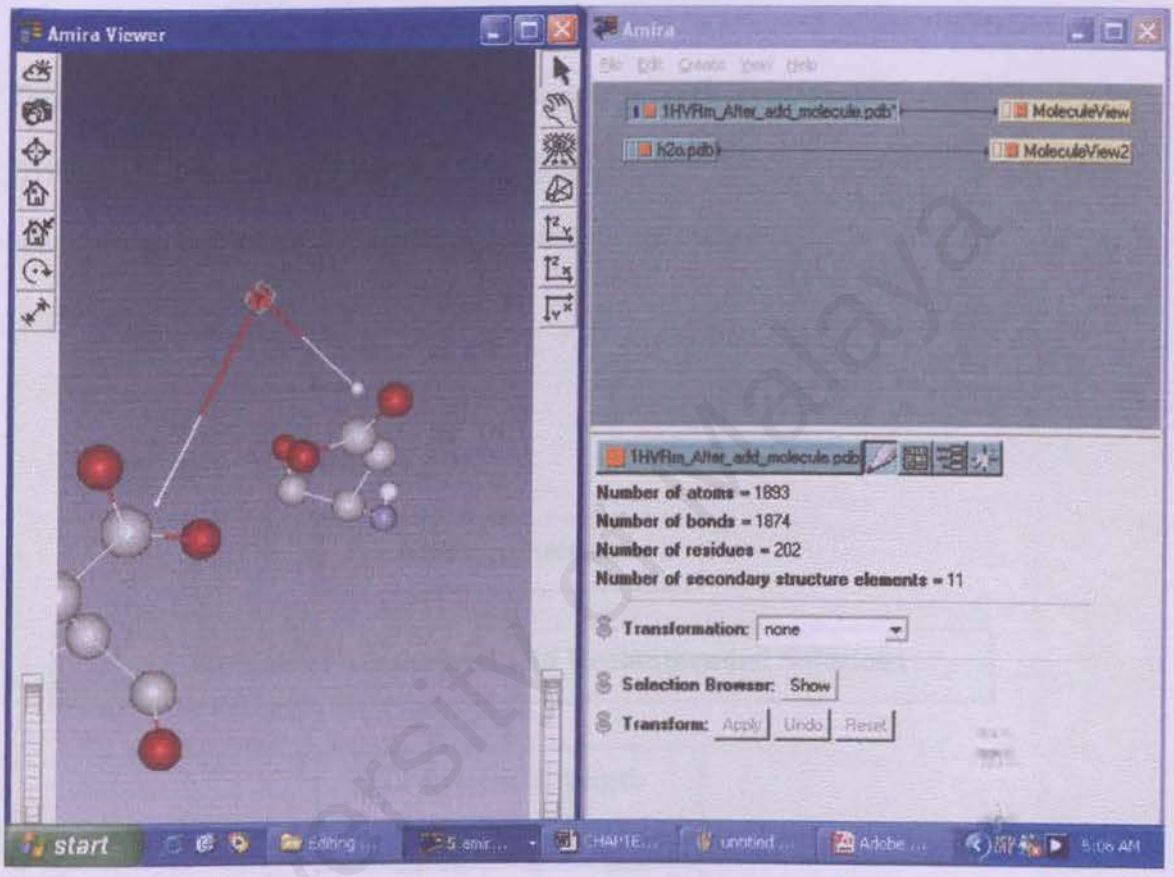
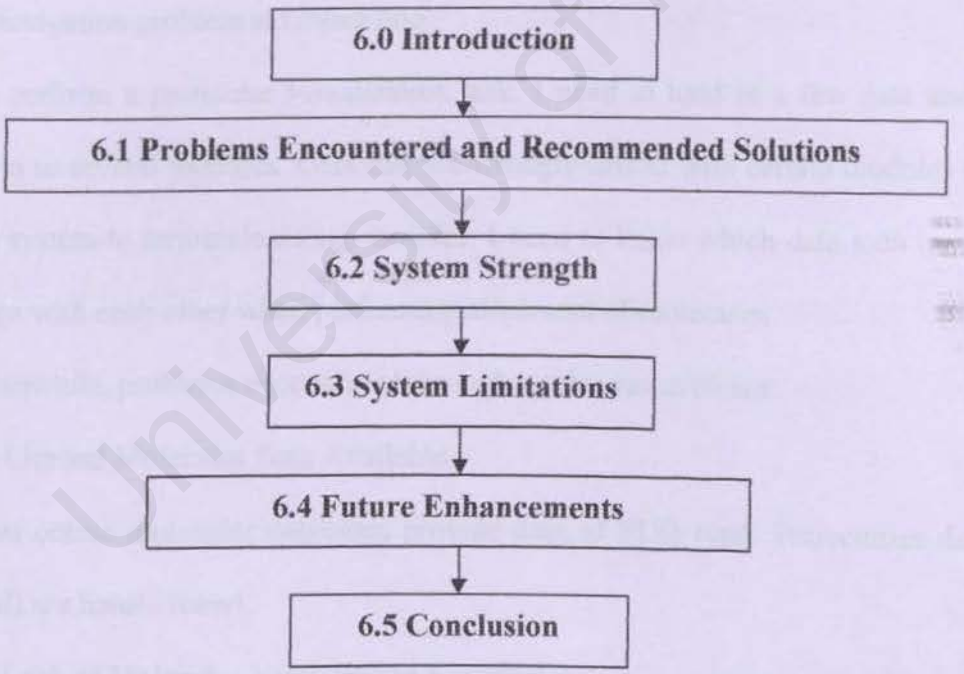


Figure 5.52: Moving dragger to its desired location

CHAPTER 6

DISCUSSION AND CONCLUSION

This chapter discussed on problems encountered and recommended solutions. It also reviews on the system strength and limitations. Lastly, it discussed on future enhancements of this system before it came to a conclusion.



6.0 Introduction

Upon completing this project, the problems encountered, system strengths and limitation are identified. Here, it outlines all the content stated above as well as listing out the future enhancement and lastly the conclusion of this project.

6.1 Problems Encountered and Recommended Solutions

Problems encountered can be divided into 2 sections, first is the problems encountered while using AmiraMol system and secondly, problems encountered while carrying out this research.

Among the problems encountered when using AmiraMol system in performing Molecule Visualization are:

i. Navigation problem at Object Pool

To perform a particular visualization task, I need to load in a few data and attached them to several modules. Data that are wrongly linked with certain modules will cause the system to terminate/debug. Besides, I need to know which data icon (molecule) to align with each other when performing alignment of molecules.

Meanwhile, problems encountered throughout this research are:

i. Limited Molecular Data Available

Most online molecular databases provide data of PDB type. Trajectories data (suffix .zmf) are hardly found.

ii. Lack of Molecular Visualization Knowledge

Having restricted knowledge in molecule and its characteristics/ attributes has been an obstacle in performing visualization. In order to do visualization, I have gathered

various ideas from online journals and past dissertations regarding molecular visualization methods and techniques, besides referring to numerous books to get an understanding of the unfamiliar terms that are frequently used in molecular biology field. The only institution that is currently implementing Amira in Molecule Visualization research is Zuse Institute Berlin, Germany. Paperworks that are published by this institution have been the main source for reference.

iii. Lack of References that Explain the Modules in AmiraMol

Amira is a newly developed 3D visualization system and therefore not much documentation or references that are available to explain the functionality of ports in each modules. I am fully dependent on Amira User's Guide to get a clear view of their working order. However, the description provided are mostly a brief one besides using terms that are only understood by expert in molecular biology field.

6.2 System Strength

i. Processing Speed

Apart from the computation of Configuration Density (with visualization using Voltex module), all other computation processes takes a reasonable duration.

ii. Consistency

This system able to generate a consistent output for each computation.

iii. Advanced Selection

This system enables users to select atoms/ residues/ chains/ secondary structures according to its name or type. This is done through Master Level in Selection Browser window, hence has simplified the segmenting task.

iv. Viewer Navigation

The viewer mode enables users to rotate the molecule to view it in different angle. Molecules can also be zoomed accordingly to get a closer look.

6.3 System Limitations

Among the system limitations of AmiraMol 3.1 are:

i. Computation of the probability density for larger sets of molecular geometries requires a much longer time.

ii. Larger molecule rendering requires a larger memory.

iii. AmiraMol unable to load in molecular data of FASTA file format (suffix .fna). Besides, it can only read and write PDB (.pdb), Tripos (.mol2) and Unichem file (.uni). Meanwhile, CHARMM file (suffix .psf and .dcd) can only be read.

iv. In MoleculeLabel module, the maximum attribute (atom's detail) that can be displayed are restricted to only 2 attributes at one time.

v. Align Sequence tool can only be implemented on proteins and nucleic acids molecules (except for t-rna molecules containing modified bases).

vi. Data that have been visualized cannot be saved as a new data. User needs to save the whole network in order to view the data that have been visualized. Only data that are modified in Molecule Editor are able to be saved as a new data.

6.4 Future Enhancements

i. AmiraMol should be designed to read in as many data file format.

ii. A better algorithm should be designed for efficient computing of probability density, such that large sets of molecular geometries can be processed at reasonable speed.

6.5 Conclusion

As stated earlier in the previous project proposal (WXES3181), the main objective of this project is to build a molecular visualization system using AmiraMol by extending its user interface. The proposed user interface should be a simplified version of the existing AmiraMol 3.1 that enables user to directly perform various visualization tasks. This interface should be able to link with the modules in the Amira system. However, due to some problems that encountered while linking the DLL (Dynamic Link Library) files of Amira has forced some changes to be made on the scope of this research. The current objective is to fully explore the functionality in AmiraMol 3.1 regarding molecular visualization.

To deeply explore AmiraMol 3.1 requires much knowledge in molecular field. By now, only the main visualization tasks have been carried out while there are still a numerous methods that are yet to explore. Not knowing the purpose of some visualization tasks, as well as the modules involved has refrained this research to be carried out much further.

However, I am glad as I managed to find out a few visualization techniques despite last minute scope changes. Having able to solve some problems faced has given me some satisfaction while carrying out this research. From this research, I have learned to

manage my time well as much time is needed to acquire skills in a field that is totally new for me. Having juggling with sudden scope changes and the limited time to deeply understand and explore each module in AmiraMol has been a challenge in this project. But, finally after all the obstacles, I've realized that the learning outcomes and the experiences gathered have proven enriching.

REFERENCES

[1] Baum, Daniel and Schmidt, Ehrenberg. "Introduction to Amira". Zuse Institute Berlin. 4 March 2003. <http://www.bcbio.de/zib_lecture/Prakt2003/lect/AmiraEinfuehrungBiomolPraktikum.pdf> (29 June 2004).

[2] Bajaj, Chandrajit and Djeu, Peter. "Interactive Visual Exploration of Large Flexible Multi-Component Molecular Complexes." 24 April 2000. <http://ccvweb.csres.utexas.edu/ccv/papers/flexible_multicomponent.pdf> (10 September 2004).

[3] Dalke, Andrew and Shculten, Claus. "Using tcl for molecular visualization and analysis." Beckman Institute. 5 August 2000. <helix-web.stanford.edu/people/altman/psb97/dalke.pd> (25 September 2004).

[4] Li, Wentian. "New Stopping Criteria for Segmenting DNA sequences." *Physical Review Letters* volume 86 , number 25 (18 June 2001).

[5] Martz,Eric. "Chime vs RasMol." University of Massachusetts. Revised 2003 <www.umass.edu/microbio/chime/chimvras.htm> (15 September 2004).

[6] Rahul. " DisMol Applet." 16 July 2001. < http://www.rahul.net/pcm/dismol/DisMol.html)> (15 September 2004).

[7] Runge, Daniel. "Algorithms and Methods for the Visualization of Molecular Surfaces and Interfaces." (Diploma diss., Institute of Computer Science, Humboldt-University of Berlin, 1999), Abstract in Digital Dissertations.

[8] Sayle, Roger A. "RasMol: Technical Introduction." Revised on 2003. <www.umass.edu/microbio/rasmol> (11 September 2004).

[9] Schach, Stephen R. *Object Oriented and Classical Software Engineering, Sixth Edition.* New York: McGraw-Hill, 2004.

[10] Schneider, Sascha and May, Thorston. "Rendering Large (Volume) Datasets: A new Parallel Visualization." 1999. <wscg.zcu.cz/wscg2003/Papers_2003/K07.pdf> (13 September 2004).

[11] TGS Release. "TGS Press Release : Features of Amira." 2002. <http://www.tgs.com/support/datasheet/amira30_Bio.pdf> (24 August 2004).

[12] Unknown. "Definitions, History, Goals, and Overview of Visualization." 15 June 2001. <http://science.kennesaw.edu/~cxu/hypervis/visgoals/visgoal0.htm> (29 August 2004).

[13] Unknown. "Introduction to DNA structure." 6 July 1998. <http://www.blc.arizona.edu/Molecular_Graphics/DNA_Structure/DNA_Tutorial.HTML > (9 August 2004).

[14] Unknown. "Surface rendering," Siggraph Online. 2000. <www.siggraph.org/education/materials/HyperVis/vis_gloss.htm> (15 July 2004).

[15] Unknown. "Volume Rendering." 14 April 1998. <www.highered.mcgraw-hill.com/sites/0072322098/student_view0/glossary_u-z.html> (15 July 2004).

[16] Unknown. "Techniques of Molecule Rendering and Representation Models." 20 March 2000. <scsg9.unige.ch/fln/eng/toc.html> (27 September 2004).

[17] Unknown. "MOLCAD: Sophisticated Graphical Rendering of Molecular Surfaces and Properties." 2002. <http://www.tripos.com/sciTech/inSilicoDisc/moleculeModeling/molcad.html> (15 September 2004).

[18] Unknown. " PSF File Format." 2002. <http://helios.bto.ed.ac.uk/evolgen/filatov/PSF.htm> (11 September 2004).

[19] Unknown. "UniChem File Format." 2002. <http://www.lrz-muenchen.de/services/software/chemie/unichem_doku/5500/5500_249.html#HEADING248> (11 September 2004).

[20] Unknown. "PDB Format." 2002. <http://www.lrz-muenchen.de/services/software/chemie/unichem_doku/5500/5500_246.html#HEADING245> (11 September 2004).

[21] Unknown. "MDL File Format." 2003. <www.graphics.cornell.edu/online/**formats/mdl**/> (11 September 2004).

[22] Unknown. " Short, non-technical Introduction to RasMol". 17 April 2002. <www.umass.edu/microbio/rasmol/> (15 September 2004).

[23] Unknown. "WebLab Viewer Version 2.0." November 1997. <http://www.msi.com/viewerlite/ > (15 September 2004).

[24] Unknown. "Chime: Browsers, Platforms, Installations, Troubleshooting." October 2002. Revised by MDL Information System Inc, March 2004. <http://www.mdli.com/chemscape/chime/> (11 September 2004).

[25] Unknown. "LITHIUM : 3D Molecular Visualization Tool." 2002. <http://www.tripos.com/sciTech/inSilicoDisc/dataAnalysis/lithium.html#3D> (15 September 2004).

[26] Unknown. "The Role of Virtual Reality in BioInformatics Applications.' 2000. <www.isima.fr/limos/publi/paper/2004/RR0407.doc> (6 August 2004).

[27] Unknown. "Visualization-terminology and citations." 1999. <www.zib.de/Visual/visual_intro.html> (29 August 2004).

[28] Unknown. "MolVis." 2002. <www.zib.de/Visual/projects/molvis/molvislong.en.html> 18 August 2004).

[29] Whitten, Jeffrey L. and Bentley, Lonnie D., *System Analysis and Design Method.* New York: McGraw-Hill, 2004.

[30] Zheng, Li. "Simultaneous Interactive Visualization of Multiple Volumes." 6 April 2000. <www.inac.purdue.edu/inac/suri2003/Li_abstract.pdf> (10 September 2004).

[31] Schmidt-Ehrenberg, Johannes. "Visualizing Dynamic Molecular Conformations." Zuse Institute Berlin (ZIB) Journal (2003).

[32] Runge, Daniel. "Algorithms and Methods for the Visualization of Molecular Surfaces and Interfaces." Diploma dissertation, Institute of Computer Science Humboldt-University of Berlin, 1999.