U18BTI7203 - BIOINFORMATICS LABORATORY

LABORATORY MANUAL & WORKBOOK

Prepared by

Dr. Ram K

DEPARTMENT OF BIOTECHNOLOGY
KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

2024-25

Contents

Experiement 1

Unix Basics

Introduction to Unix Commands

Objectives

- Apply basic Unix commands to perform essential file and directory operations such as creating, modifying, and deleting files and directories.
- Execute simple shell scripts to automate repetitive tasks, demonstrating an understanding of basic scripting concepts in a Unix environment.

1.1 Install Windows Subsystem (WSL) in Window

Official Microsoft Installation - WSL Installation

1.2 Unix Command Structure

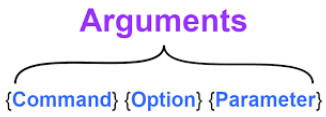

Figure 1.1: 3-Piece Anatomy of Linux Commands

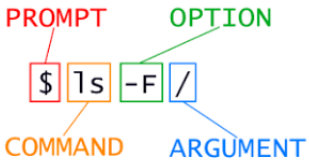

Figure 1.2: Sample command for navigating files

1.3 Getting Started with Unix Commands

Activities

1. **Getting Started with Unix Terminal:**

   Open the terminal and write a command to display the current directory.

   ```
   pwd
   ```

   List the files and directories in the current directory.

   ```
   ls
   ```

1.4 Basic File Operations

Objectives

- Learn to create, view, and delete files and directories.

Activities

1. **File Creation and Deletion:**

Create a new directory named "Bioinformatics"

```
mkdir Bioinformatics
```

Create a file inside this directory.

```
touch Bioinformatics/sample.txt
```

Delete the file.

```
rm Bioinformatics/sample.txt
```

1.5 Text Processing Commands

Objectives

- Understand how to view and manipulate text files.

Activities

1. **Viewing and Editing Files:**

   Display the contents of a file.

   ```
   cat filename.txt
   ```

   Use `grep` to find lines containing a specific word.

   ```
   grep "word" filename.txt
   ```

1.6 File Permissions and Ownership

Objectives

- Learn how to view and change file permissions.

Activities

1. **Changing Permissions:**

   View the permissions of a file.

   ```
   ls -l filename.txt
   ```

   Change the permissions to make the file executable.

   ```
   chmod +x filename.txt
   ```

1.7 Basic Scripting

Activities

1. **Writing and Running a Script:**

   ```
   echo 'echo
   "Hello, Bioinformatics!"' > hello.sh
   chmod +x hello.sh
   ./hello.sh
   ```

   Create a simple script that prints "Hello, Bioinformatics!".

1.8 Practical Bioinformatics Examples

Objectives

- Apply Unix commands and scripting to bioinformatics data.

Activities

1. **Working with Sequence Data:**

   ```
   grep -c ">" sequences.fasta
   ```

   Download a sample sequence file (e.g., FASTA format) and count the number of sequences in the file.

1.9 Cheatsheet

Basic Commands

- `pwd` - Print working directory
- `ls` - List directory contents
- `cd [directory]` - Change directory
- `mkdir [directory]` - Create new directory
- `rmdir [directory]` - Remove directory
- `touch [file]` - Create new empty file
- `rm [file]` - Remove file
- `cp [source] [destination]` - Copy file
- `mv [source] [destination]` - Move/rename file
- `who` - Who all have logged into unix
- `who am i` - Who and when was logged
- `man` - help page

Date Operation

- `date` - Prints the current date
- date + %D - mm/dd/yy
- date + %H - Hr -00 to 23
- date + %M - Min 00 to 60

File Operations

- `cat [file]` - Display file contents
- `more [file]` - View file with pagination
- `less [file]` - View file with navigation
- `head [file]` - Display first 10 lines of file
- `tail [file]` - Display last 10 lines of file
- `grep "pattern" [file]` - Search for pattern in file
- `find [directory] -name "[name]"` - Find files by name

Permissions

- `ls -l` - List files with permissions
- `chmod [permissions] [file]` - Change file permissions
- `chown [owner]:[group] [file]` - Change file owner and group
- `chgrp [group] [file]` - Change file group

Text Processing

- `echo "text"` - Print text to terminal
- `wc [file]` - Word, line, character count
- `sort [file]` - Sort file contents
- `uniq [file]` - Remove duplicate lines
- `cut -d[delimiter] -f[field] [file]` - Extract fields
- `paste [file1] [file2]` - Merge files line by line

Networking

- `ping [host]` - Ping a host
- `ifconfig` - Display network configuration
- `netstat` - Network statistics
- `scp [source] [destination]` - Secure copy over SSH
- `ssh [user]@[host]` - SSH to remote host

System Information

- `uname -a` - Kernel information
- `top` - Task manager
- `df -h` - Disk space usage
- `du -sh [directory]` - Directory size
- `free -h` - Memory usage
- `uptime` - System uptime

Compression

- `tar -cvf [archive.tar] [files]` - Create tar archive
- `tar -xvf [archive.tar]` - Extract tar archive
- `gzip [file]` - Compress file
- `gunzip [file.gz]` - Decompress file
- `zip [archive.zip] [files]` - Create zip archive
- `unzip [archive.zip]` - Extract zip archive

Scripting Basics

- `echo "Hello, World!"` - Print text
- `VAR=value` - Define variable
- `$VAR` - Access variable
- `if [ condition ]; then ... fi` - Conditional statements
- `for VAR in [list]; do ... done` - Loop
- `while [ condition ]; do ... done` - While loop

Miscellaneous

- `alias ll='ls -la'` - Create alias
- `history` - Command history
- `man [command]` - Show manual for command
- `which [command]` - Show command path
- `ps` - List processes
- `kill [pid]` - Kill process by PID

1.10 Practice Questions

Instructions

Question 1 (3 marks)

Analyze the directory structure and file contents in a given directory. Write a command or a series of commands to:

1. List all files in the current directory and its subdirectories.
2. Find all files that contain the word "bioinformatics" and display their names.
3. Count the number of lines in each file that contains the word "sequence".

Question 2 (2 marks)

Examine file permissions in a directory. Write a command or a series of commands to:

1. Change the permissions of all files in the current directory to read-only for the owner, and no permissions for others.
2. List the files to verify the changes.

Question 3 (3 marks)

You have a large log file that contains several entries with the format: "date time log_level message". Write a command or a series of commands to:

1. Extract all entries with the log level "ERROR".
2. Sort these entries by date and time.
3. Save the sorted entries to a new file named "error_logs.txt".

Question 4 (3 marks)

Create a directory called 'dept'. Create subdirectories called fin, pers, mktg under this directory. Create two subdirectories under fin called exp and inc. Create two subdirectories under pers directory called admin and pay. Under the exp directory create two files called ent and postage. Create two files fees and sales under the inc directory. Create two files exp and reco under the admin directory. Create two files undee the pay directory called ded and add. Create files called salesinfo and pur under the mktg directory. Answer the following question based on this.

1. Copy ded to the mktg directory. Rename ita s deduc and create a link file for it. No move the link file to admn directory

2. Copy saleinfo to pay directory and move it to the exp directory under the name sinfo. Copy the same file in the fin directory and remove the file sinfo in exp directory
3. Rename the file postage to pexp and move ti to the pers directory. Now create link for the same file and remove the pexp file
4. Print the tree using the package tree. Install the package if necessary.

Question 4 (2 marks)

Analyze a given dataset stored in a CSV file. Write a command or a series of commands to:

1. Extract the second and fourth columns.
2. Remove duplicate rows based on these columns.
3. Save the unique rows to a new file named "unique_data.csv".

***

Experiement 2

Biological Sequence Analysis

Objectives

- Develop Critical Analysis Skills: Students will enhance their ability to critically analyze and compare sequence data from different bioinformatics databases, identifying key differences in metadata, sequence accuracy, and functional annotations.
- Understand Impact of Database Management: Students will gain a comprehensive understanding of the implications of data redundancy, database updates, and user interface design on sequence retrieval and bioinformatics research, enabling them to make informed decisions in their future research projects.

2.1 Basics of Sequence Search

2.1.1 Keyword Searches on GenBank

The default GenBank page is a search page which allows us to search for sequences using various keywords.

Figure 2.1: Basic Search using Genbank

In the left side of the search we can select the database which we want to search (e.g. nucleotide, protein, genome). Also, in the keyword searches, logical arguments like `AND, OR, NOT` can be used together various categories. The categories are mentioned in square brackets. e.g. [Organism].

Example:

```
Rodentia[Organism] AND 12S rRNA[Gene Name] NOT Mus[Organism]
```

When defining names for various organisms, please make sure the organisms queried are spelled exactly as in GenBank's taxonomic database or use their respective unique taxon ID (for Rodentia it is txid9989[Organism]).

An advantage of this approach is that the result can be easily filtered to complete sequences, sequence type (mRNA, genomic, protein) or taxonomic tree.

2.2 Searches Using Sequence Homology (BLAST – Basic Local Alignment Search Tool) on GenBank.

BLAST-Link A disadvantage of keyword search is that genes annotated differently from the query are not retrieved. Also, using keywords can be quite laborious, for example if you want to download sequences from non-model taxon representatives. In these cases we use homology searches, such as BLAST. Basic types of BLAST are:

1. BLASTN (Nucleotide BLAST) – the query is a nucleotide sequence, searched against a nucleotide database.
2. BLASTP (Protein BLAST) – the query is a protein sequence, searched against a protein database.
3. BLASTX – the query is a nucleotide sequence which will be translated in six frames, searched against a protein database (takes longer because it represents multiple searches per query).
4. TBLASTN – the query is a protein sequence, searched against a nucleotide database translated into protein space.
5. TBLASTX - the query is a nucleotide sequence which will be translated in six frames, searched against nucleotide databases translated into protein (takes longer).

Choose one of these approaches on the main page or change your choice using the tabs of the application page. Then enter your query sequence(s) into the large field.

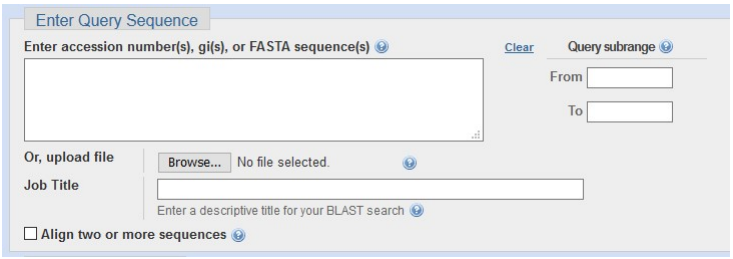

Figure 2.2: Search using BLAST

`For BLASTX/TBLASTX, remember to select the correct Genetic code for your query.`

In the section `Choose Search Set`, select a target database. If not selecting human or mouse, you can enter optional database parameters, such as taxon limit or other criteria.

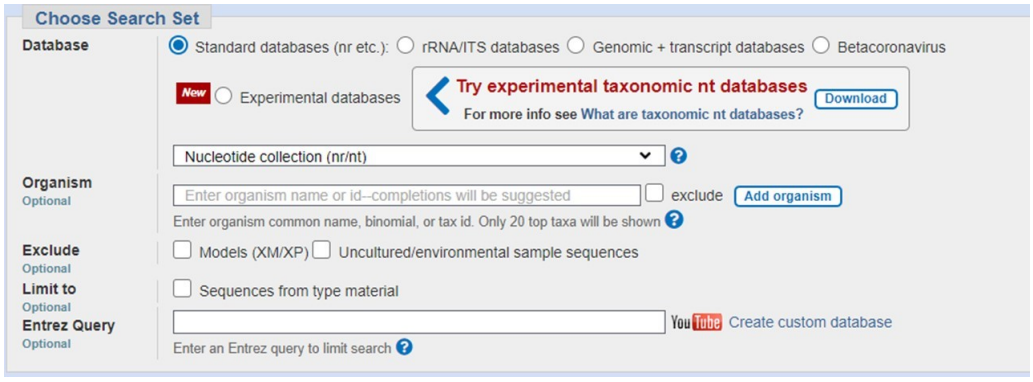

Figure 2.3: Search Set

In the section `Program Selection`, choose a subtype of algorithm based on expected sequence similarity and start your search by clicking the “BLAST” button.

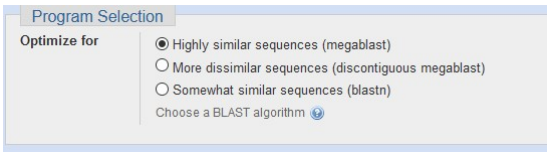

Figure 2.4: Search Set

A disadvantage of this method is that sequences are retrieved based on similarity, not function. Therefore, a subsequent phylogenetic analysis is often helpful in resolving sequence homologies, for example within gene families.

2.2.1 Sequence Formats

FASTA - plain-text, simple format, usually the input format for many sequence analysis programs.

```
>seqname1 description of sequence
AACTGCCATGCAACCATGACTAGCATA
```

GenBank - metadata-rich format, shown by default for sequences in GenBank, e.g. Example

2.2.2 Downloading sequences

Sequences from GenBank can be downloaded one by one or in batch.

- Downloading sequences one by one
    - Let's try this with a chloroplast-encoded gene from *Euglena gracilis*
    - Using Copy Ctrl(Cmd) + C and Paste Ctrl(Cmd) + V, we can copy the sequence from the webpage and paste it in Notepad++ (gedit in Ubuntu, BBedit in MacOS). Afterwards we can save the file as .fasta.
- Another way to save the sequence is the top right menu Send to:. From here we can choose to save the entire sequence or just the coding sequences, and we can select various file formats.
    - After selecting the format we click on Create File and we save the file.
- Batch downloading of the sequences
    - Both keyword and sequence similarity search approaches allow you to download the results as sequence batches.
    - For the keyword approach: Select the sequences of interest using the checkbox on the left (or check none to download all). Using the Send to: button, define the parameters of the output.
    - For BLAST hits: On the result page, select the sequences of interest using the checkbox or using the Select all button. Clicking the Download starts the export process by letting you choose the output format.

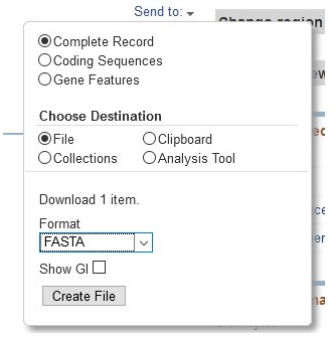

Figure 2.5: Downloading Sequence from Genbank

In the top left corner of each sequence record there is a link called FASTA. Clicking on the link will show the entire sequence in FASTA format.

2.2.3 Sequence editing

For basic editing of your sequence datafiles you can use freeware editors, e.g. UGENE/AliView. We will mainly focus on AliView.

- AliView basics
  - Input formats – FASTA, Phylip, Nexus and others (not GenBank unfortunately)
  - Double-click a sequence name to view/edit it. Check sequence statistics using the graph tab on the right.
  - Ctrl(Cmd) + A to select all.
  - Toggle translation using a button (see if sequences are in frame):
- Aliview repos: Aliview Repos

Figure 2.6: Aliview. Download link Download here

Instructions

- Answer all questions thoroughly.
- Use appropriate bioinformatics tools and databases as necessary.
- Provide screenshots or output snippets where applicable.
- Submit your completed worksheet by [Due Date].

Question 1: Comparative Analysis of Database Structures

Task: Compare the structure and features of three major sequence databases: GenBank, EMBL-EBI, and DDBJ.

Instructions

1. Retrieve a nucleotide sequence of the same gene (e.g., *BRCA1*) from GenBank, EMBL-EBI, and DDBJ.
2. Compare the metadata fields provided by each database for the same sequence.
3. Evaluate the ease of use and the comprehensiveness of each database.

Questions

1. What are the key differences in the metadata fields provided by each database?
2. Which database provides the most comprehensive information for the selected gene?
3. How does the user interface of each database affect the retrieval process?

Question 2: Sequence Retrieval Accuracy

Task: Evaluate the accuracy and consistency of sequence retrieval from multiple sources.

Instructions

1. Retrieve the protein sequence for the human hemoglobin subunit beta (*HBB*) from UniProt, NCBI Protein, and Ensembl.
2. Align the sequences retrieved from each database.
3. Analyze any discrepancies among the sequences.

Questions

1. Are there any differences in the sequences retrieved from the three databases? If yes, describe them.
2. What might be the reasons for any discrepancies found?
3. Which database would you consider the most reliable for protein sequence retrieval and why?

Question 3: Evaluating Data Redundancy

Task: Assess the issue of data redundancy in sequence databases and its implications.

Instructions

1. Select a bacterial genome (e.g., *Escherichia coli*) and retrieve its sequence from GenBank.
2. Identify and list all instances of sequence redundancy within this genome entry.
3. Discuss the impact of data redundancy on sequence analysis and research.

Questions

1. How many redundant sequences did you find within the genome entry?
2. What are the potential problems caused by data redundancy in sequence databases?
3. Suggest methods to minimize redundancy in sequence databases.

Question 4: Functional Annotation Consistency

Task: Compare functional annotations for a specific protein across different databases.

Instructions

1. Choose a protein (e.g., *TP53*) and retrieve its functional annotations from UniProt, InterPro, and Pfam.
2. Compare the annotations provided by each database.
3. Evaluate the consistency and reliability of the functional annotations.

Questions

1. What are the similarities and differences in the functional annotations provided by each database?
2. Which database provides the most detailed functional annotation for the protein?
3. How do discrepancies in functional annotations affect downstream bioinformatics analyses?

Question 5: Impact of Database Updates

Task: Analyze the effect of database updates on sequence retrieval and research.

Instructions

1. Retrieve a sequence entry for a well-studied gene (e.g., *CFTR*) from a database (e.g., GenBank) at two different points in time (e.g., current date and a date from two years ago).
2. Compare the two entries to identify any changes or updates.
3. Discuss the potential impact of these changes on ongoing research.

Questions

1. What changes or updates were observed in the sequence entries retrieved at different times?
2. How can such updates affect the results and interpretations of bioinformatics research?
3. What strategies can researchers use to stay updated with the latest sequence information?

Questions 6: BLAST Search

1. You have obtained a sequence for an unknown human gene, and you suspect it to encode a protein. We will try to find out which protein this is using BLASTX. Click here to download seq input [Ref ID: seq1]
2. 1. From the protist *Paratrimastix pyriformis* we obtained the following protein. We will look for homologous sequences in the database and assess its function by using the distantly related protein search tool PSI-BLAST (Position-Specific Iterated BLAST). Click here to download seq input[Ref ID: Seq2]

Question 7: BLAST Search Advanced

1. Download all available enolase sequences from the phylum Parabasalidea. Use a sequence homology search of BLAST and retrieve one sequence per species. As query, use the *Trichomonas vaginalis enolase* sequence: Click here to download the seq input[Ref ID: Seq3]

Submission Guidelines

- Compile your answers into a single document.
- Include any necessary figures, screenshots, or code snippets.
- Ensure all references to external sources are properly cited.

***

Experiement 3

Molecular Visualization using Pymol

Objectives

- Enhance Structural Comprehension: Students will develop a thorough understanding of the three-dimensional structures of proteins, DNA, and RNA, enabling them to visualize and interpret the spatial arrangements and interactions of these macromolecules.
- Apply Visualization Tools Effectively: Students will gain proficiency in using molecular visualization tools and software, allowing them to create, manipulate, and analyze detailed models of macromolecules for advanced bioinformatics and structural biology research.

3.1 Basics of Molecular Visualization

3.1.1 Getting started with PDB

PDB files are essentially a set of atomic coordinates or "atom placements" read by PyMOL in a defined format to allow a molecular structure to be built. PDB file names consist of a number followed by three characters – for example, **1HEW**. Although it does not follow common convention, in this tutorial capitals are used for the letters of PDB file codes, primarily because many file names contain the number 1 or the letter L, which are identical in most fonts if not capitalized. (The number zero sometimes occurs in PDB file names, so watch out for the subtle difference between 0 [zero] and O [the letter O]. Depending on your browser font, these might be completely indistinguishable.)

3.1.2 Downloading the PDB

You will start learning about PyMOL by looking at the enzyme Hen Egg lysozyme in complex with the trisaccharide inhibitor tri-(N-acetylglucosamine) or tri-NAG (see a biochemistry text if you want to know more about lysozyme). Atomic coordinates for this model are contained in the Protein Data Bank file with code 1HEW. You can obtain this file from the Protein Data Bank using your web browser by following these instructions:

1. Go to www.rcsb.org and open a new window for the PDB Home Page. You may wish to add this site to your menu of bookmarks or favourites
2. A search box with a Enter search terms(s) is near the top of the screen, in the middle. Type 1HEW (the first character is the number one) in the box. Then click the Site Search (magnifying glass) button. The Structure page for 1HEW appears.
3. At the right side of the page, click Download Files to reveal a drop-down list, and then click PDB format.
4. Your browser should download the file, perhaps asking you where to save it under the file name 1HEW.pdb. Remember where you saved the 1HEW.pdb file as you may have to open it from within PyMOL using the File > Open option from the pull-down menu

The typical pymol workspace is listed

Figure 3.1: RCSB PDB Databank retrieval process

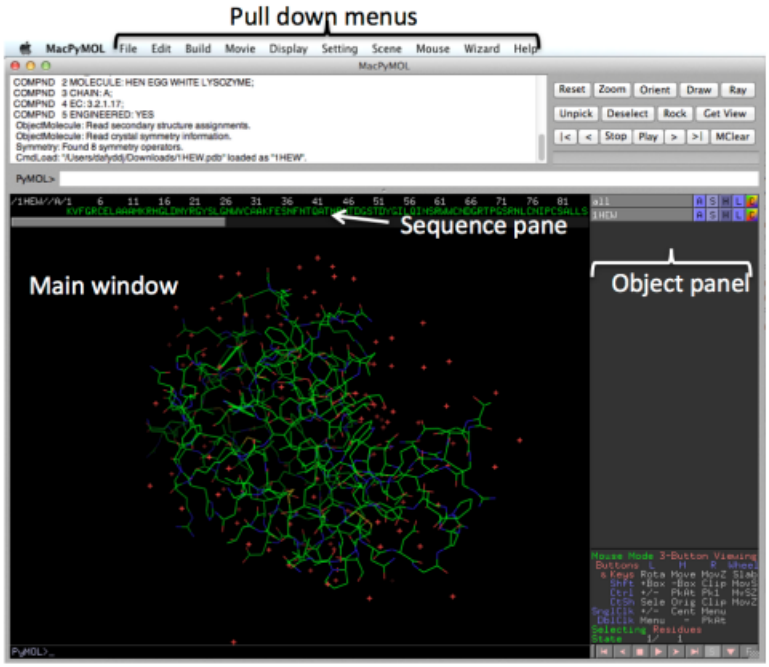

Figure 3.2: Pymol main menu

3.1.3 Pymol Commands

```
# load the 3 pdb files and execte the following commands
pymol mol1.pdb mol2.pdb mol3.pdb -d 'as ribbon;spectrum count;set seq_view'

# load the 3 pdbs and run the script
pymol mol1.pdb mol2.pdb mol3.pdb my_script.pml

# Run my_program using PyMOL as the Python interpreter
# passing the 3 pdb files to my_program.py
pymol my_program.py -- mol1.pdb mol2.pdb mol3.pdb
```

3.1.4 Mouse Option

```
Rotate:       Click & Drag
XY-Translate: Option-Click & Dreg
Zoom:         Control-Click & Drag
Select:       Click & Release
Box-Select:   Shift-Click & Drag
Box-Deselect: Shift-Option-Click & Drag
Clipping:    Control-Shift-Click & Drag with...
* near plane controlled by vertical motion
* far plane controlled by horizontal motion
```

3.1.5 Structural Representation

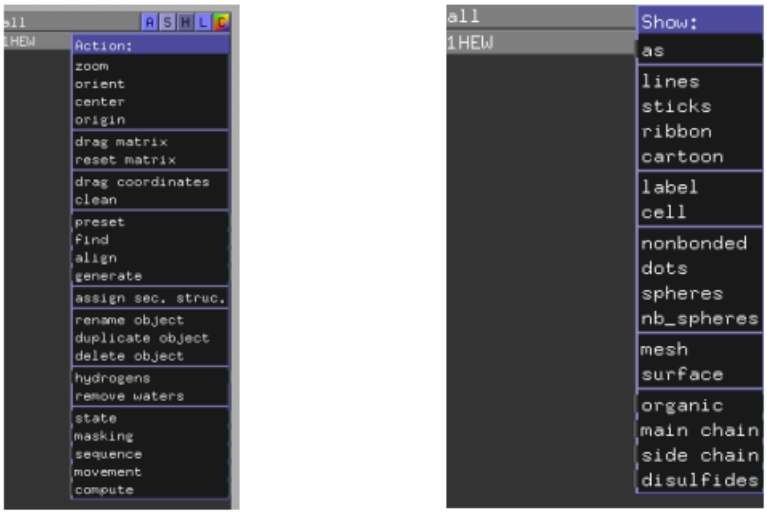

Figure 3.3: Structural Representation

3.1.6 Selection Algebra

PyMOL's selection language allows to select atoms based on identifiers and properties. Many commands (like color, show, etc.) take an atom selection argument to only operate on a subset of all atoms in the scene. Example:

```
PyMOL>show spheres, solvent and chain A
```

Selections can be made more precise or inclusive by combining them with logical operators, including the boolean and, or, and not. The boolean and selects only those items that have both (or all) of the named properties, and the boolean or selects items that have either (or any) of them.

3.1.6.1 Selection Operator/Modifier Table

More info on the selection algebra can be found in - https://pymolwiki.org/index.php/Selection_Algebra

3.1.7 Selection Macros

Selection Macros allow to represent a long atom selection phrase such as

```
PyMOL> select pept and segi lig and chain B and resi 142 and name CA
```

in a more compact form:

```
PyMOL> select /pept/lig/B/142/CA
```

3.1.8 Property Selectors

PyMOL reads data files written in PDB, MOL/SDF, Macromodel, ChemPy Model, and Tinker XYZ formats. Some of the data fields in these formats allow PyMOL to assign properties to atoms. You can group and select atoms according to these properties using property selectors and identifiers: the selectors correspond to the fields in the data files, and the identifiers correspond to the target words to match, or the target numbers to compare.

General : The items in a list of identifiers are separated by plus signs (+) only. Do not add spaces within a list of identifiers. The selector resi takes (+)-separated lists of identifiers, as in

```
PyMOL> select nterm, resi 1+2+3
alternatively
PyMOL> select nterm, resi 1-3
The identifier for a blank field in an input file is an empty pair of quotes
PyMOL> select unstruct, ss ""
```

More details on the property selector can be found in - https://pymolwiki.org/index.php/Property_Selectors#Property_Selector_Table

3.1.9 Color

color sets the color of an object or an atom selection to a predefined, named color, an RGB hex color, or a color ramp. For an overview of predefined colors, see Color Values. For a script that enumerates all the colors see, List_Colors. If you want to define your own colors, see Set_Color.

Usage:

```
color color-name
color color-name, object-name
color color-name, (selection}
```

Examples - Color all carbons yellow

```
color yellow, (name C*)
```

Example - Color by element, except carbons

```
color atomic, (not elem C)
```

3.1.10 Coloring secondary structures

To assign helices, sheets and loops individual colors, do:

```
color red, ss h
color yellow, ss s
color green, ss l+''
```

When the colour bleeds from the ends of helices and sheets into loops, do:

```
set cartoon_discrete_colors, 1
```

Or activate Cartoon -> Discrete Colors in the GUI menu.

3.1.11 Assign color by B-factor

B-factor coloring can be done with the spectrum command. Example:

```
spectrum b, blue_white_red, minimum=20, maximum=50
as cartoon
cartoon putty
```

3.1.12 Selecting secondary structures

```
select helix, (ss h)
select sheet, (ss s)
select loop, (ss l+'')
```

3.1.13 Manually Assigning Secondary Structure

You can manually assign secondary stuctures to your protein by

```
alter 96-103/, ss='S'
alter 96-103/, ss='H'
alter 96-103/, ss='L'
```

to set residues 96-103 to beta Strand, alpha Helix, and Loop respectively.

3.2 Sample Workflow

1. **Load a Structure:**

   ```
   load 1UBQ.pdb
   ```

2. **Select and Highlight Specific Regions:**

   ```
   select alpha_helix, resi 1-20
   show cartoon, alpha_helix
   color red, alpha_helix
   ```

3. **Generate a High-Quality Image:**

   ```
   ray
   png alpha_helix.png
   ```

3.3 Worksheet

Analyzing Molecular Structures Using PyMOL

1. **Load a PDB File:**
    - Load the structure of Hemoglobin (PDB ID: 1A3N) into PyMOL.
2. **Task 1: Identifying Secondary Structures**
    - Select and color all alpha-helices in red.
    - Select and color all beta-sheets in blue.
    - Save the session and export the image.
3. **Task 2: Visualizing Ligand Binding**
    - Identify the ligand in the Hemoglobin structure.
    - Create a selection for the ligand and color it yellow.
    - Show the ligand in stick representation.
    - Save an image showing the ligand binding site.
4. **Task 3: Mutations and Structural Changes**
    - Introduce a mutation at a specific residue (e.g., mutate residue 58 from Valine to Glycine).
    - Observe and describe the structural changes.
    - Save a session file with the mutation.
5. **Task 4: Generating a Report**
    - Generate a report (in a text file) that includes:
        · Commands used for each task.
        · Images saved for each task.
        · A brief description of the observations for each task.

***