Interim Report

Vinay Kakkar

December 6, 2022

Contents

1 Aims, objectives and Literature Survey . . . 1
1.1 Aims . . . 1
1.2 Objectives . . . 1
1.3 Futher Objectives . . . 2
2 Protein Structures . . . 3
2.1 Introduction . . . 3
2.2 Primary, Secondary, Tertiary and Quaternary Structure . . . 3
2.3 Large Scale Experssion . . . 6
2.4 RNAseq . . . 9
2.5 Alpha Fold . . . 10
3 The Protein Data Bank and the File Formats . . . 10
3.1 Protein Data Bank . . . 10
3.2 PDB Currently . . . 13
3.3 Recent Improvements . . . 14
3.4 Summary . . . 15
3.5 File Formats . . . 16
4 Hadoop spark and pyspark . . . 18
5 Software Development . . . 20
5.1 Unit Tests . . . 20
5.2 Branching . . . 21
6 Planning and Time-Scale . . . 21
6.1 Planning . . . 22
6.2 Time-Scale . . . 22
7 Diary . . . 22

Abstract

Currently, biologists are analysing PDB files that contain structural protein data in which they can determine something from running a user executable on to it. Biologists can determine many things from the analysis of these types of data. Biologists are then able to aid in new drug discoveries as a result help create new medicine for mutated proteins that might not be in living cells at the moment. This project aims to help that process in being able to gather lots of data together and make it efficient for a biologist to run an executable onto these files. To do this I delve into protein structures in the PDB bank and Hadoop/spark which provides batch processing features that allow me to distribute these files onto a cluster that is then able to run in parallel. This is my interim report in which I explain my proof of concept programs to show that this framework is possible and will help the structural bioinformatics department in analysing thousands of data in a very quick manner.

1 Aims, objectives and Literature Survey

1.1 Aims

Aim provided in the Project description

Provide a framework where large numbers of protein structures can be analysed using a user-provided executable in the MapReduce formalism.

Aim in more depth

When a biologist conducts analysis on a protein structure they can learn about the functionality and behaviour of the protein. With this aim it is very beneficial for the biology sector as this will help them perform large scale data analysis on to protein structures. This is very helpful as it allows biologists to be able to learn from the analysis about protein structures functionality and behaviours in a more efficient manor. As a result, this will allow biologists to be able to create a new drug which can help with future or current diseases called new drug discovery.

By using batch processing and large-scale data processing will give the biologist the ability to do so. This means that given lots of protein structures the user would be able to pass each protein structure of their choice into a executable of their choice that will return a result. As a result, being able to perform more analysis on more protein structures yielding faster results. More specifically we can use Hadoop or Spark with python to store these protein data and process the data in parallel so that multiple proteins can be processed at the same time giving better speeds and performance yielding faster results.

For this to be possible we need to build a program that functions as a framework that has multiple user executables and that can be flexible in adding in newer created executables in the future which can be picked by the user when deciding what analysis, they want to conduct. essentially this program should be able to handle the protein data provided by the protein data bank and most the file types provided by it. It is important that the user should be able to choose what protein data it wants to send into the framework and what executable they want to conduct on these files.

1.2 Objectives

To achieve the aims provided in section 1 i will need to complete a set of objectives which once finished will suggest that the aim has be met and that I fulfil the needs for the aims.

ObjectivesCompleted

Tutorials

In order to complete the projects aim i need to understand some technologies:

- Spark
- Pythons API for Spark pyspark
- How to create a cluster
- How to perform Mapreduce
- How to set up a local cluster

I went through a few tutorials which are named in my readme file within GitLab.

The first two tutorials showed me how to set up pyspark and how to use spark and python together. This was mainly to do with the setup and basics of spark.

The next tutorial taught me how to conduct MapReduce within pyspark on spark dataframes. Focusing more on complicated features such as MapReduce functions of spark and how they work.

Lastly, the last tutorial showed me how to distribute these dataframes on a cluster (local) by turning the dataframes into an RDD which can be spread across the nodes of a cluster.

I tried to follow a tutorial that showed me the basics of setting up a cluster among computers but this quickly escalated and did not fit within the time frame and schedule I had planned.

Proof of Concept

Before beginning my program I try to achieve the needs of my aim. To do this I need to create a program that proves that it is possible. The two main things I need to prove are that I can set up a cluster and distribute a single file amongst it. It is important to note I use the correct file type which will relate to the same type I would use in the actual program. The second thing I need to prove is that I can run a simple executable onto this file which is distributed. As if I can do both features then it shows that my actual program will indeed work and I wont run into any unfixable blocks along the way.

Distributing Protein Structure Files Amongst MapReduce cluster

First action to take is create a local cluster. Then we need to read a .pdb file or multiple in a directory and convert it into a spark dataframe which is then converted into a rdd so that it can be distributed on to the nodes in the cluster(in this case it will be how every many cores i give my local cluster).

Running MapReduce using a single type of executable for Analysing Protein Structures

This is completed by collecting all the data from an rdd back into dataframes and then pack each datafram value into a PDB file which we can then run user executables on to yield a result.

Reports

Whilst working on the proof of concept programs I also need to research the data I am going to be working with.

Protein Structures

Researching this topic gave me in-depth knowledge of protein structures and large-scale expressions for example different methods used on how the data is extracted to be fed into the file. This gave me insight into the files I am working with and also aided me to navigate through the PDB website as I understood some of the hard biological language used within the site.

The Protein Data Bank and the file formats

The protein data bank provides the data in fields that can be of three different types; .pdb, .mmcif, .XML

This is important as it will be the file types I will be working with for example I will need to put some of these file types into a cluster whilst working with pyspark. This gave me insight into how to read these files and made it easier to create my proof of concept programs as I understood results when trying to read the file or turning the file into a dataframe.

1.3 Futher Objectives

The Objectives provided above are completed but together do not satisfy the needs of the aims there are some objectives left that i will need to complete to complete each aim listed these include:

- Provide some form of UI so that a user can pick what executables to perform and also be able to upload what protein structures they want to provide analysis on.
- Be able to perform/use user executables that are perfomred on pdb files which will yeild a result (Currently my proof of concept program runs a terminal command that returns the number of lines present in the .pdb file).
- Reasearch on how to distribute clusters to multiple computers allowing them to comunicate and perform actions and achive goals needed by working together.
- Reasearch on how to conduct benchmarking comparing the speeds difference when trying to accomplish the same task but not using batch processing.

- Provide a clear set of guidelines for the map/reduce executables.

These objectives will form the basis for my second term work and so will my time scale be created from these.

Extensions

If given the change/time i would work on some extra features that will improve my program. Some extensions i have thought of are:

- Use a very big data set as a test. For example, this data set can be a subset mirror of the pdb site. This can yield intresting results in testing the performance of my framework.
- Improved ui as biologists are not to deep in the computer science sector having a easy ui can really help and enahnce there experiance whilst using this program.
- A set of user executables that are ready and just need to be selected by the user.
- Implementation on a Public Mapreduce cluster e.g. Amazon.

literature survey

2 Protein Structures

2.1 Introduction

Amino acids are molecules that when combined forms proteins. All of the 20 amino acids, see table 1 have in common a central carbon atom which is attached to a hydrogen atom, an amino group, and a carboxyl group. What distinguishes one amino acid from another is the side chain attached to the central carbon atom through its fourth valence [BT98].

Proteins are responsible for catalysing most of the chemical reactions in cells. They can function as enzymes catalysing a wide variety of reactions important for life and thus also important for the structure of living systems such as proteins involved in the cytoskeleton. The size of protein can vary [Zve08].

Definition 2.1 (Catalysing) *Catalysing is to make a chemical reaction happen or happen more quickly by acting as a catalyst.*

Definition 2.2 (Cytoskeleton) *A dynamic network of interlinking protein filaments present in the cytoplasm of all cells [Zve08].*

We can analyse a DNA sequence of a gene to retrieve the amino acid sequence of the protein product, using the fact that proteins are built up of amino acids. Leaving a position where we can help deduce the likely properties of unknown proteins, whilst at the same time including their functions and structures. Knowing the relationship between a protein's structure and its function provides a better understanding of how the protein works with better understand this we can conduct experiments to explore how modifying the structure will affect the function. The use of bioinformatics aids this process whilst also providing computer modeling for these interactions [Zve08].

2.2 Primary, Secondary, Tertiary and Quaternary Structure

Please refer to 1 for a visual representation.

Primary Structure

The primary structure of a peptide or protein is the linear sequence of its amino acids. It is read and written from the amino-terminal to the carboxyl-terminal end. Where each amino acid is connected to the next by a peptide bond. Primary structure sequence it can interact with one another to form secondary structures [SFB04].

Amino acid	Three-letter code	One-letter code
Glycine	Gly	G
Alanine	Ala	A
Valine	Val	V
Leucine	Leu	L
Isoleucine	Ile	I
Proline	Pro	P
Phenylalanine	Phe	F
Methionine	Met	M
Tryptophan	Trp	W
Cysteine	Cys	C
Asparagine	Asn	N
Glutamine	Gln	Q
Serine	Ser	S
Threonine	Thr	T
Tyrosine	Tyr	Y
Aspartic acid	Asp	D
Glutamic acid	Glu	E
Histidine	His	H
Lysine	Lys	K
Arginine	Arg	R

Table 1: The 20 amino acids. The amino acid name, the three-letter code, and the one-letter code are given. The Amino acids are split up into Nonpolar, Polar, Acidic and Basic respectfully

Secondary Structure

The secondary structure refers to the local arrangement of a peptide chain. Where several common secondary structures have been identified in proteins [SFB04].

Tertiary Structure

Tertiary structure is a three-dimensional structure of a protein the formation is built up of bonds and interactions that serve to change the shape of the overall protein. Finaly the folding that we end up with for a given polypeptide is the tertiary structure [God22].

Quaternary Structure

The quaternary structure of a protein is built-up of several protein chains/subunits. Each of the subunits has its primary, secondary, and tertiary structure. The subunits are held together by hydrogen bonds and van der Waals forces between nonpolar side chains [OR15]. I highlighted some proteins with Quaternary structures 2.

Definition 2.3 (Van Der Waals) *A relatively weak electric force attracts neutral molecules that collide with or pass very close to each other [Bou18].*

Protein	Number of Subunits	Function
Alcohol dehydrogenase	4	Enzymatic reaction in fermentation
Aldolase	4	Enzymatic reaction in glycolysis
Fumarase	4	Enzymatic reaction in citric acid cycle
Hemoglobin	14	Oxygen transport in blood
Insulin	2	6344

Table 2: Examples of Proteins Having Quaternary Structure [OR15].

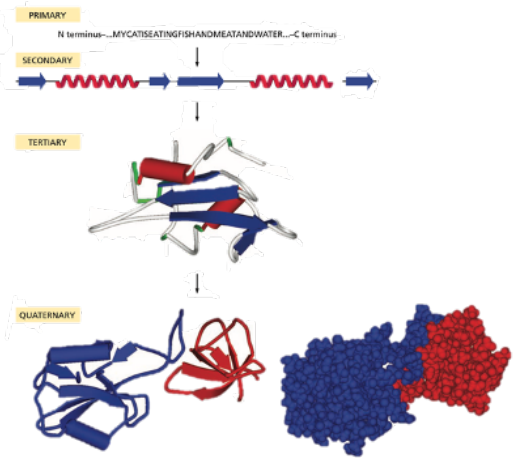


Figure 1: From the sequence alone, the primary structure to secondary structure, to tertiary structure(3D), to finally quaternary structure found when several tertiary structures form a multisubunit complex [Zve08].

Considering Protein structure on several different levels

The fold of the protein plays part in determining the way the protein will function, and also whether it will function correctly so it is important to understand these folds refer to 1 to see what a protein fold looks like. Which we can use to help us for example predict the fold of a protein from its sequence. Looking at Protein structures on different levels we need to consider the analysis of protein structure by experimental techniques such as X-ray crystallography, nuclear magnetic resonance, and RNAseq which show that proteins adopt distinct structural elements [Zve08].

Amino Acids

When looking at a primary structure of a protein 1 the sequence of amino acids 1 will build up the linear protein chain. This linear chain is often called a polypeptide chain [Zve08].

Amino acids are different from each other due to their side chains and due to this the functional properties of various different proteins are different. Each type of amino acid has specific chemical-physical properties determined by the structure and chemical properties of its side chain. They can, however, be classified into overlapping groups that share some common physical and chemical properties [Zve08]. You can see the amino acids grouped here 1.

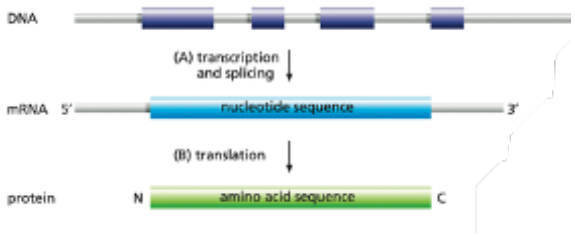


Figure 2: The relation of DNA coding-strand sequence to mRNA sequence to protein sequence. The exons of the DNA are transcribed into mRNA which, using other molecules directs the protein sequence [Zve08].

Bioinformatic Difficulties with Predictions on Proteins

It is difficult to define the precise ends of the helices(The secondary structure of proteins is made up of a-helices and b-strands) for structures found in globular proteins that are not perfectly regular. Making it one step more difficult when trying to predict these structures [Zve08].

To Note:

- Several different types of b-sheet are found in protein structures.
- Turns, hairpins, and loops connect helices and strands.
- Any chain between two regular structures is referred to as a loop.
- Mostly a loop will contain a turn (or even several).

In antibody recognition, immunoglobulins employ loops at the edge of a b-sheet. All immunoglobulin structures with the same overall chain fold, but it is the difference at these loops that results in different results. Loops take up one of a limited number of structures called canonical forms. This type of classification is another reason why trying to predict both the structure and function of the protein is difficult [Zve08].

Definition 2.4 (Immunoglobulin) *Immunoglobulins are heterodimeric proteins composed of two heavy and two light chains. Types of white blood cells that helps the body fight infection [SC10].*

2.3 Large Scale Experssion

Gene expression begins when genes are transcribed into messenger RNAs, which are then translated to produce proteins.

Total gene expression in cultured cells or a tissue sample can be detected in three main ways:

1. DNA microarray technology.
2. Two-dimensional Gel electrophoresis or Chromatography.
3. RNAseq

With the first two cases, they produce enormous amounts of raw data [Zve08] due to this, many proteins currently evade high-resolution structure determination. Structural mass spectrometry is a powerful approach that is better than the first two methods mentioned above, by having nearly an unlimited size constraint and speed. Although the data provided by mass spectrometry is vague for full high-resolution structure elucidation, structural mass spectrometry can be used to examine the size, solvent accessibility, and topography of proteins [LLV18] [LZG20]. Many mass spectrometry techniques exist that can elucidate elements of protein tertiary and quaternary structure [BL22].

We can have computational methods that aid experimental technique intending to elucidate protein structures [SL20] [LWL+20]. Software packages can be used to combine data with advanced structure sampling and scoring techniques. Computational tools for protein structure modeling, include the Rosetta software suite [LWL+20] [ALFJ+17], I-TASSER [YYR+15], Phyre2 [KMY+15], Integrative Modeling Platform [RLW+12], HADDOCK [DBB03], and MODELLER [EWMR+06] [BL22].

Large Scale Gene Expression

Genome DNA microarray experiments produce large amount of data can be computationaly heavy on where methods can yield alternative conclusions from inceasing the computational effort.

The goal of these experiments is to determine biological or functional meaning from the lists of genes, either by:

1. Identify critical genes that are responsible for a biological effect.
2. Find patterns within the genes that point to an underlying biological process.

[Zve08]

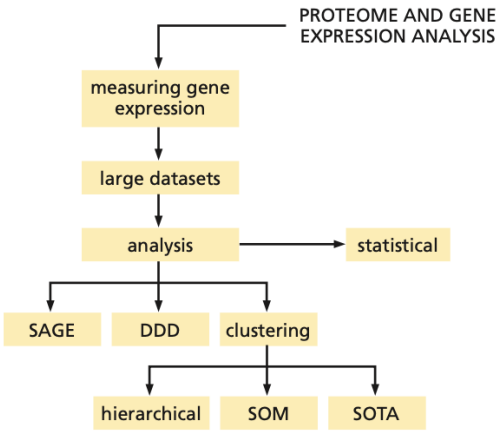


Figure 3: Describing Common experimental aspects of gene expression and of the analysis of the resulting data [Zve08].

Serial analysis of gene expression

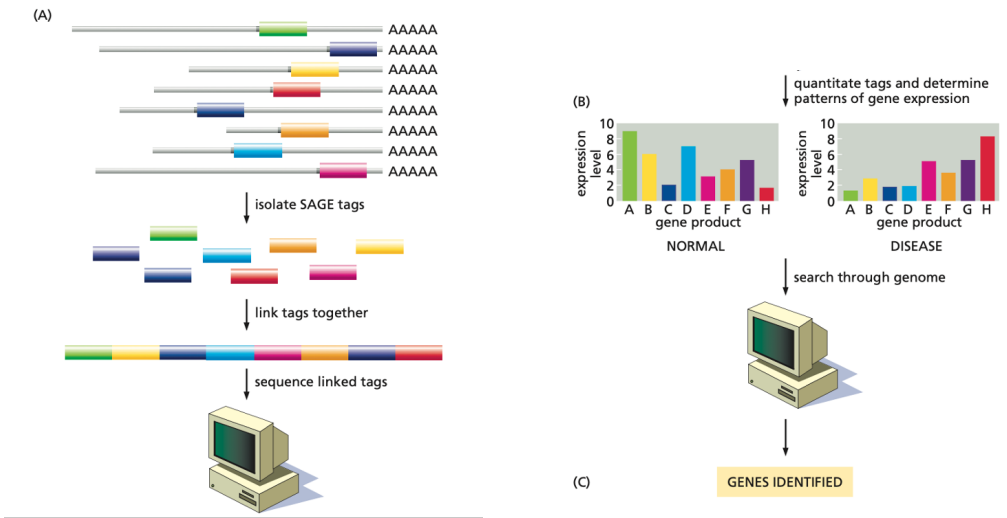
Serial analysis of gene expression is the alternative compared to microarrays when trying to investigate patterns of gene expression.

A short sequence contains enough information to uniquely identify a gene. The sequence tags from the total cellular RNA can be linked together to form long DNA molecules. The total number of times a particular tag is observed the concatemers approximates the expression level of the corresponding gene. The data produced by SAGE include a list of the tags with their corresponding counts, providing a digital output of cellular gene expression. Which allows the user to specify which organ is to be investigated. Libraries consisting of gene lists organized by the various types of tissues or cell lines are provided for further choice. The output from SAGE provides the SAGE tag, the UniGene ID, the gene description, and color and letter-coded differences in expression levels [Zve08].

Clustered gene expression data

Clustered pattern data obtained from gene expression microarrays/genome bioinformatics can be used as a tool to identify new transcription factors or other cell-regulatory proteins.

The clustered genes/proteins can be analyzed. Leading to a vast collection of data from many gene/protein expression experiments being available on the Web [Zve08].



(a) (b)

Figure 4: *An outline of the SAGE method for comparing levels of gene expression. (A) Short sequence tags. The sequence tags are isolated and are linked together to produce long DNA molecules that can be cloned and sequenced. (B) Once sequenced, each tag can be calculated, resulting in a value that gives the expression level of the corresponding transcript [Zve08].*

Large Scale Protein Expression

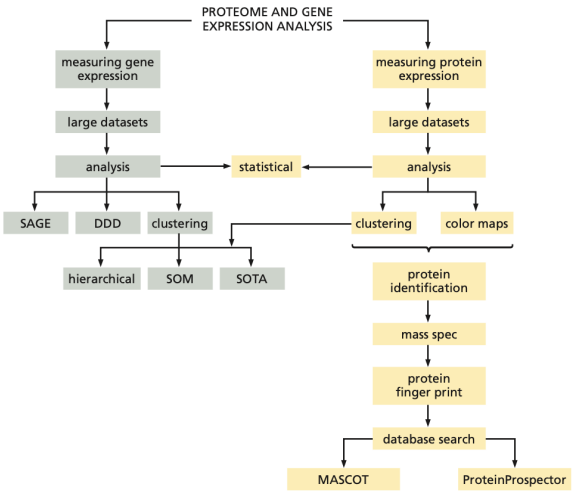


Figure 5: Describing some experimental aspects of protein expression and of the analysis of the resulting data. [Zve08].

For functional protein, mRNAs need to be translated, whilst the protein products can change which influence their function. For this reason we can measure and anlayse different proteins.

There is more proteins than there are genes in a genome. Transcripts can be spliced in various ways to give different mRNAs, providing different protein products, from the same gene. However, proteins that can be modified after translation giving more different protien products.

Protien expressions can vary in a organism depending on the origin and it will also differ between the separate stages of an organism's life cycle and under different environmental conditions.

Definition 2.5 (proteome) *The proteome refers to all the proteins that make up an organism at a specific point in time and under specific conditions.*

It is important to know how protein expression is affected in order to understand how an organism or a cell functions [Zve08].

2.4 RNAseq

The transcriptome is important for revealing the molecular constituents of cells and tissues, interpreting the functional aspects of the genome, also for understanding development and disease [WGS09].

Many methods deduce and quantify the transcriptome, including hybridization or sequence-based approaches. For example, hybridization-based approaches involve incubating fluorescently labeled cDNA with microarrays or commercial high-density oligo microarrays [WGS09].

However, these methods have several limitations, such as:

- Dreliance upon existing knowledge about genome sequence.
- Limited dynamic range of detection owing to both background.
- High background levels owing to cross-hybridization [OM06] [RRG07].
- saturation of signals.

Definition 2.6 (transcriptome) *The transcriptome is the complete set of transcripts in a cell, and their quantity, for a specific developmental stage or physiological condition.*

Sequence-based approaches directly determine the cDNA sequence such as Tag-based methods which include SAGE, CAGE [KKN+06], MPSS [RBL+02].

Each approach is high throughput and can provide precise, gene expression levels. However, a significant portion of the short tags can not be uniquely mapped to the reference genome [WGS09].

RNA-Seq RNA sequencing has clear advantages over existing approaches it uses deep sequencing technologies where a population of RNA is converted to a library of cDNA fragments with adaptors attached to one or both ends. Each molecule is then sequenced in a high-throughput manner to obtain short sequences from one or both ends [WGS09].

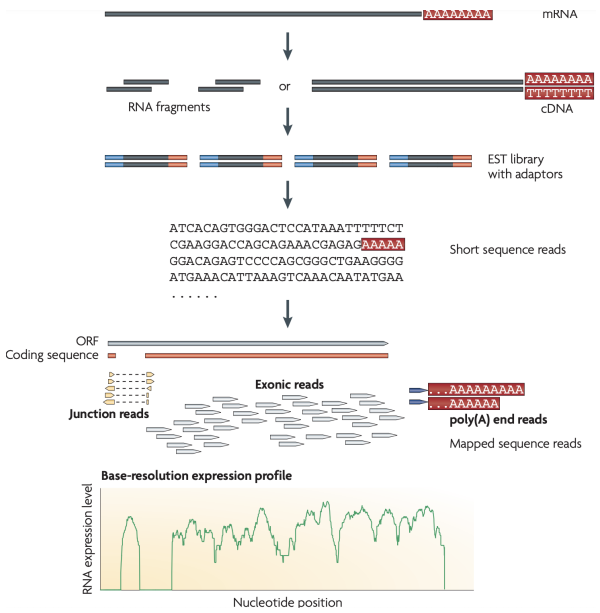


Figure 6: A typical RnA-seq experiment. RNAs are first converted into a library of cDNA fragments through either RNA fragmentation or DNA fragmentation. Sequencing adaptors are subsequently added to each cDNA fragment and a short sequence is obtained from each cDNA using high-throughput sequencing technology. The resulting sequence reads are aligned with the reference genome or transcriptome.

2.5 Alpha Fold

AlphaFolds' goal is to predict the 3D coordinates of all heavy atoms for a given protein using the primary amino acid sequence and aligned sequences of homologues as inputs [JEP+21].

Mutations in proteins can lead to misfolding which is often associated with disease states, for example, Alzheimer's and Parkinson's which is one of the challenges for alphaFold [Fel].

The output is a file containing the 3D coordinates for every non-hydrogen atom in the protein. whilst showing the confidence levels for every amino acid residue, providing the reliability of the predicted structure [Fel].

Bioinformatics with Alpha Fold

In July 2021, AlphaFold was developed by DeepMind and was made available to the public [TAW+21].

Where it tries to solve the issue of invariant protein structures that are under translations and rotations [BP03].

AlphaFold is trained on protein chains from the PDB using the input sequence to query databases of protein sequences to generate a multiple sequence alignment [JEP+21]. Although we still do not exactly know how a protein sequence folds and alpha fold do not help in figuring this out its impact will likely be in accelerating and improving the production of new medications [NZLJ22].

AlphaFold 2

The CASP14 was recently held which is a blind trial that critically assesses techniques for protein structure prediction [DITS22], AlphaFold2 was entered and out-performed all competitors.

Recently, RoseTTAFold was developed, trying to implement similar principles. Since then, other end-to-end structure predictors have emerged using different principles such as fast multiple sequence alignment processing in DMPFold218 and language model representations.[BPE22].

We use the root mean square deviation, to calculate the similarity between the two structures, AlphaFold models had an accuracy of 0.96 compared to 2.80 which was the second-best score. AlphaFold models also had a high level of accuracy in predicting the position of residue side chains when the protein backbone prediction was accurate [DITS22] [JEP+21].

3 The Protein Data Bank and the File Formats

3.1 Protein Data Bank

The Protein Data Bank was established at Brookhaven National Laboratories [BKW+77] in 1971 as an archive for biological macromolecular crystal structures [BWF+00].

Definition 3.1 (Macromolecular) *Macromolecular is any very large molecule, usually with a diameter ranging from about 100 to 10,000 angstroms*

It is an information source for data retrieved from atomic structures, crystallography, and three-dimensional structures of biomolecules, including nucleic acids and proteins [BG21].

At the time this was the first open-access digital data resource in biology which started with just seven protein structures [BBB+22b].

Various groups such as the Protein Data Bank in Europe, Protein Data Bank Japan help manage the Protein Data Bank archive. Current wwPDB members also include the ElectronMicroscopy Data Bank and the Biological Magnetic Resonance Bank [BBB+22b].

Protein Data Bank China has recently joined the wwPDB as an Associate Member with its role as wwPDBdesignated PDB Archive Keeper. Where they are responsible for weekly updates of the archive and safeguarding both digital information and a physical archive of correspondence [BBB+22a].

The management of PDB must comply with FAIR (the acronym depicts: Findable, Accessible, Interoperable, Reusable) and FACT [vdABH17] guiding principles for scientific data [WDA+16] [WSHB20].

The FAIR Guiding Principles	
To be Findable:	F1. (meta)data are assigned a globally unique and persistent identifier F2. data are described with rich metadata (defined by R1 below) F3. metadata clearly and explicitly include the identifier of the data it describes F4. (meta)data are registered or indexed in a searchable resource
To be Accessible:	A1. (meta)data are retrievable by their identifier using a standardized communications protoco A1.1 the protocol is open, free, and universally implementable A1.2 the protocol allows for an authentication and authorization procedure, where necessary A2. metadata are accessible, even when the data are no longer available
To be Interoperable:	I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation. I2. (meta)data use vocabularies that follow FAIR principles I3. (meta)data include qualified references to other (meta)data
To be Reusable:	R1. meta(data) are richly described with a plurality of accurate and relevant attributes R1.1. (meta)data are released with a clear and accessible data usage license R1.2. (meta)data are associated with detailed provenance R1.3. (meta)data meet domain-relevant community standards

Table 3: The guidlines to what builds up the FAIR principles [WDA+16]

Aims and Objectives of PDB

Enzymology, electron microscopy, computational chemistry small molecule crystallography, biochemistry, biophysics, macromolecular crystallography and nuclear magnetic resonance spectrometry all help the aims and goals of the PDB archive [BG21].

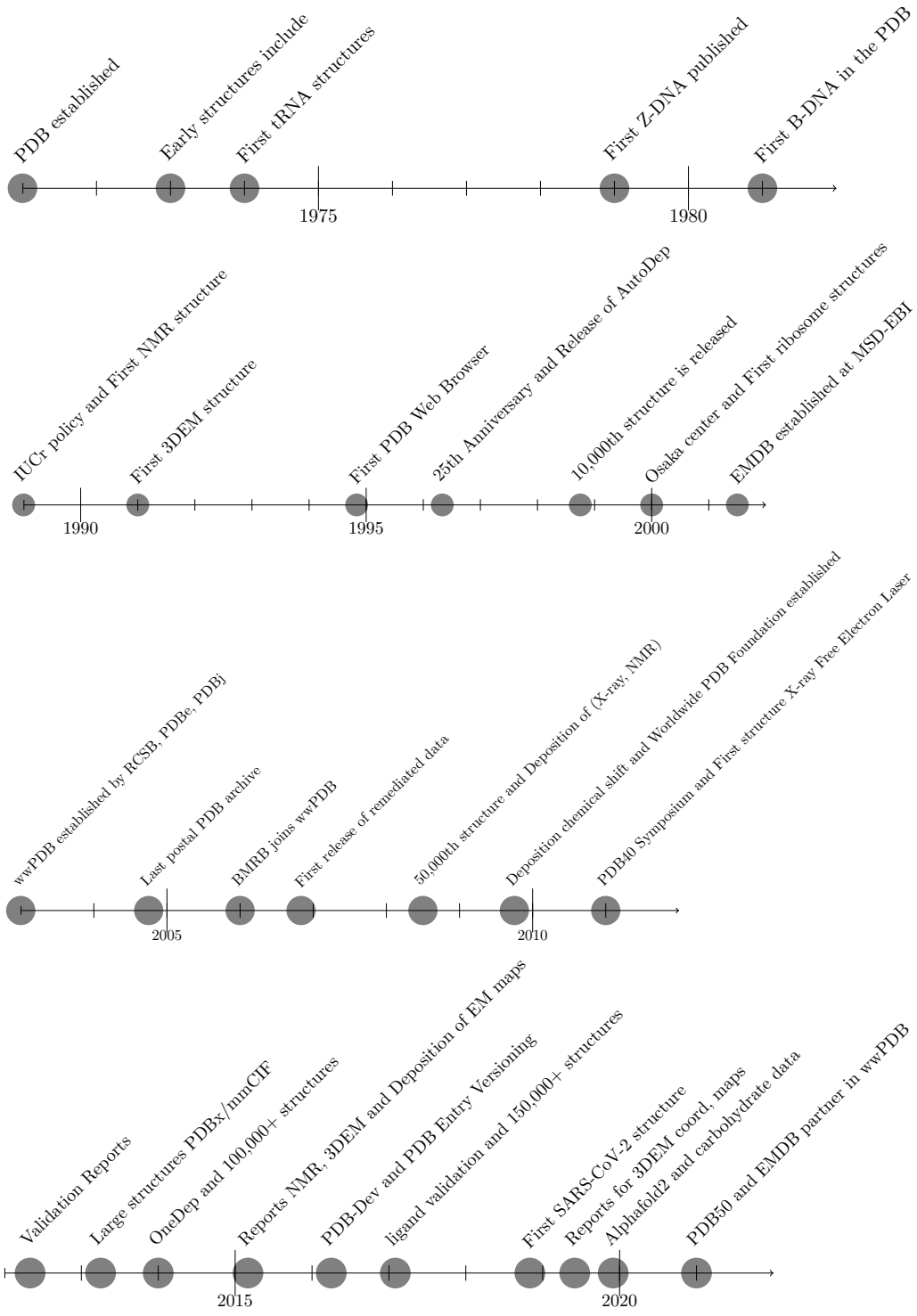
Definition 3.2 (Enzymology) *Enzymology is the branch of biochemistry aiming to understand how enzymes work*

Definition 3.3 (Electron Microscopy) *Electron microscopy is a technique for obtaining high resolution images of biological and non-biological specimens.*

PDBs provide open access to nearly 200 000 archived, validated, and biocurated experimentally determined three-dimensional structures of biological macromolecules.3D structures archived in the PDB have enabled important scientific breakthroughs by basic and applied researchers [Bur21]. Open access to PDB data without restrictions on usage has also aided structural bioinformatics in areas such as computational biology.

Timeline of PDB

I have provided a timeline representing the milesstones achived within the protein data bank. Where PDB marked its 52st anniversary of continuous operations.



Recent Project

A project was undertaken to change the information management services for RCSB.org. The idea was to have developed a primary place for studying 3D biostructures by extending RCSB.org web portal functionality to support parallel delivery of more than one million CSMs publicly available from AlphaFold DB and ModelArchive together [BBB+22a].

Covid

During the COVID-19 pandemic, more than 2000 structures associated with the agent of the coronavirus disease were released and have become accessible to global users for free. The properties of these structures give us this opportunity to find out the ligand binding sites, the spatial conformation of ligands, protein-to-protein interactions, and amino acid substitutions regarding different viral proteins. Moreover, chemical, functional and energetic characteristics can also be gained to describe the potential capabilities of each molecule. These properties might aid us to determine the potential drug targets for drug design and vaccine preparation [LZD+20].

3.2 PDB Currently

As of 2022, the PDB has a vast number of 3D biostructures, eukaryotic protein structures exceeded 105 000. Bacterial protein structures were also numerous, totaling nearly 66 000. Archaeal protein structures were the least numerous totaling 5500. However the PDB coverage is decidedly limited, with mouse protein structures being most numerous at 8000 structures [BB21].

We have powerful tools developed by RCSB PDB for searching and analysis which include structure, sequence, sequence motif, structure motif, and visualization [BBB+22a].

Upon reaching the RCSB.org home page, users can query, organize, visualize, analyse, compare, and explore PDB structures and CSMs side-by-side. Searching 3D structure information can encompass PDB structures and CSMs or be limited to PDB structures only. Either PDB structures or CSMs can be excluded from the search results. The two types of structure information accessible via RCSB.org are clearly distinguished from each other. Top bar searching and data delivery for PDB structures and CSMs [BBB+22a].

The PDB Site currently

Figure 7: Search options at RCSB.org include Top Bar or Basic Search; Advanced Search; and Browse Annotations [BBB+22a].

(a) (b)

Figure 8: *Top Bar or Basic Search options available from every RCSB.org web page. Examples of searching for 3D structures using (A) simple text string insulin receptor; (B) drop down autosuggestions based on the text string insulin receptor; (C) Boolean operators to combine insulin + receptor (+ = AND); or (D) an amino acid sequence. (E) Searching RCSB.org documentation using a text string biological assembly [BBB+22a]*

3.3 Recent Improvements

Recent RCSB PDB data architecture improvements

In 2020, RCSB PDB had an upgrade of its delivery architecture [RDL+21]at RCSB.org [BBB+21]. The legacy monolithic data delivery application was changed into a distributed deployment of individual microservices, each with a single responsibility. Data access services provide both Representational State Transfer and GraphQL API access to a data warehouse hosted in a MongoDB document-oriented database. Originally, advanced Search QueryBuilder functionality encompassed text, PDB data attributes, 3D structure, sequence, biopolymer sequence motif, and chemical similarity. Every search function is implemented as an independent service. A separate search service is responsible for launching each search function, combining and delivering their integrated results to public programmatic search APIs. When each service has a single responsibility, we have greater flexibility in scaling the deployment of services in response to changes in user load and significant reductions in the time required to develop, test, and deploy new features. The Sequence Motif search function has been extended with a new 3D Structure Motifsearch capability [BBR20].

Recent advances in RCSB PDB data integration

RCSB PDB integrates the content of each expertly biocurated Entry with information from more than 50 external data resources.

Integrated external data needs to follow a data schema that defines the organization of the RCSB PDB data warehouse. Finally, it is available to RCSB PDB front-end services, public data access APIs, and our text search indexing service [BBB+22b].

Recent PDBx/mmCIF data standard improvements

The PDBx/mmCIF data standard is maintained by the wwPDB organization in collaboration with wwPDBPDBx/mmCIF Working Group domain experts recruited from the scientific community. The PDBx/mmCIF web resource supports browse and search access to standard terminology. The Working Group includes developers for many of the widely used structure determination software systems, who ensure that data produced by these programs comply with the PDBx/mmCIF data standard, generating complete and correct data files for PDB deposition. The wwPDB and the Working Group collaborate on developing terminologies for new and rapidly evolving methodologies such as Free Electron Laser, 3DEM, Serial Crystallography, and X-ray, whilst improving representations for existing data content. Most recently, the Working Group has focused on modernizing content descriptions for processed X-ray diffraction data, including extensions describing anisotropic diffraction limits, unmerged reflection data, and new quality metrics of anomalous diffraction data. Deposition and delivery

External Resources	
AlphaFold DB	Computed Structure Models by AlphaFold2
ATC	Anatomical Therapeutic Chemical (ATC) Classification System from World Health Organization
Binding MOAD	Binding affinities
Binding DB	Binding affinities
BMRB	BMRB-to-PDB mappings
Cambridge structural Database	Crystallographic small molecule data from the Cambridge Crystallographic Data Centre
CATH	Protein structure classification- Class, Architecture, Topology/fold, and Homologous superfamily
ChEMBL	Manually curated database of bioactive molecules with drug-like properties
CSD	Cambridge Structural Database: Validated and curated small-molecule organic and metal-organic crystal structures from the Cambridge Crystallographic Data Centre
DrugBank	Drug and drug target data
ECOD	Evolutionary Classification of Protein Domains
EMDB	3DEM density maps and associated metadata
ExplorEnz	IUBMB Enzyme nomenclature and classification
Gencode	Human and Mouse Gene annotations

Table 4: Some of the External Resources Integrated Into RCSB PDB

improve our ability to assess experimental data quality, and every PDB data consumer's ability to Find and Reuse relevant PDB Entries [BBB+22b].

3.4 Summary

Future and struggles of PDB

Future

As the PDB archive has started its 52nd year, it gives open access to analyses of structures and much more to: basic and applied researchers, educators, and students spanning fundamental biology, biomedicine, bioenergy, bioengineering, and biotechnology, with key points that help many communities that use this facility. Firstly It delivers Data In and Data Out services efficiently to a user base that is now numbering many millions worldwide. Secondly, it has wwPDB partners that process, validate, and biocurate the growing number of increasingly complex PDB depositions received. Manages and safeguards the growing PDB archive in its role as wwPDB designated Archive Keeper. Thirdly it enables searching, visualization, exploration, and analysis of experimentally-determined PDB structures integrated with more than one million CSMs through its web portal. [BBB+22a].

Struggles

Even after all the advancments PDB has gone through there are still additional challenges lying ahead which include:

- Rapid growth in public-domain CSMs of individual polypeptide chains, already numbering ¿200 million at the time of writing.
- Anticipated advances in AI/ML-based prediction of structures of multi-protein complexes.
- Continued development of biomolecular structure determination methods using X-ray Free Electron Lasers, revealing the microscopic details of chemical reactions in real time.
- Growth in the number and complexity of atomic-level cryoelectrontomography structures of macromolecular machines.
- Integration of PDB structures and CSMs with complementary information coming from correlative light microscopy and related imaging methods across length scales ranging from atoms to small molecules to individual biomolecules to macromolecular assemblies to organelles to cells and ultimately tissues
- Merging of the PDB-Dev prototype archiving system for integrative methods structures with the PDB archive
- Federating other biodata resources, such as the SmallAngle Scattering Database and the Proteomics Identification Database, with the PDB, EMDB and BMRB core archives jointly managed by the wwPDB partnership

[BBB+22a].

3.5 File Formats

The PDB archive holds a few different types of file types that hold data such as atomic coordinates and other information describing proteins and other biological macromolecules. Depending on what the data is created from it can fall into a different category.

PDB Data

The main information in the PDB archive is coordinate files for biological molecules. These files list the atoms in each protein and their 3D coordinates.

These files are available in several formats:

- PDB
- mmCIF
- XML

The header section of the text summarizes the protein, citation information, and the details of the structure solution, which is then followed by the sequence and a long list of the atoms and their coordinates. It also contains the experimental observations used to determine atomic coordinates [Goo].

.pdb Files

The PB format consists of a collection of records that describe the atomic coordinates, chemical and biochemical features, and experimental details of the structure determination [WF03].

Each item of data in the PDB format is assigned to a one of PDB record types (HEADER. SOURCE. REMARK, etc.). The ATOM records the atomic coordinate data [WF03].

PDB format has been extended with new REMARK records. For example, REMARK 3 that encodes refinement information has been modified and extended for each new refinement program and program version [WF03].

The PB format uses fixed-width fields to represent data, so we have limits on the size of certain items of data. For example, we cant have more then 99,999 atoms and polymer chain can be only one character. This means some structures are devided into multiple files [WF03].

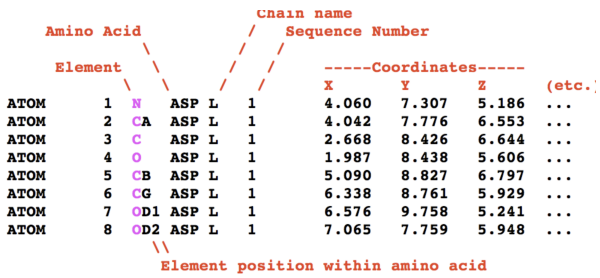


Figure 9: Showing contents of a PDB file for the Atom values [AAB+19].

.mmCIF Files

Mmcif is a dictionary-based approach to data extracted from crystallographic experiments [WF03].

It includes all the data we can find in a pdb file. Also, we have sufficient data names so that the experimental section of a structure paper can be written automatically and to facilitate the development of tools i.e. computer programs could easily access and validate mmCIF data files [WF03].

.xml

XML builds from a PDB Exchange dictionary. Although presented in very different syntaxes, the PDB Exchange and XML representations use the same logical data organization. [WIN+05].

The dictionary data block is mapped to the standard top-level XML schema element, and the data file data block is mapped to a datablock element. Category or table definitions in the Exchange dictionary are described as XML complex types. The category definition. [WIN+05].

PDB Exchange data dictionary attributes	XML schema mapping
Data block	Root level *schema element*
Category groups	
Categories	*complexTypes*
Definition	*annotation* and *documentation* elements
Examples	*annotation* and *documentation* elements
Primary keys	*attributes* of the data category
Items	*elements* of the data category
Definition	*annotation* and *documentation* elements
Examples	annotation and documentation elements
Data types	*simpleTypes*
Range restrictions and allowed values	*restrictions* within *simpleTypes* or *unions* of *simpleTypes*
Mandatory data code	Element attributes *minOccurs* and *nillable*
Parent-child relationships	*key/keyref* elements
Interdependency/exclusivity	
Units of measurement	Additional *fixed attributes*
Subcategory membership	

Figure 10: Summary of the correspondences between PDB Exchange data dictionary and XML schema metadata [WIN+05].

Visualizing Structures

PDB files can be viewed from text editors but we can also use a browsing or visualization program. RCSB PDB allows you to search and explore the information, including information on experimental methods and the chemistry and biology of the protein. Visualization programs allow to read of the

PDB file and, display the protein structure generating custom pictures of it. These programs can contain analysis tools that allow you to measure distances and bond angles, and identify interesting structural features [Goo].

Reading Coordinate Files

Before exploring structures in the PDB archive we need some prior understanding of the coordinate files. For example, we can find a diverse mixture of biological molecules, small molecules, ions, and water which can get confusing we can use the names and chain IDs to help sort these out. In structures determined from crystallography, atoms are annotated with temperature factors that describe their vibration and occupancies that show if they are seen in several conformations. NMR structures often include several different models of the molecule [Goo].

Potential Challanges

There are some things to note as you could fall into some challenges when browsing through the PDB archive. Many structures, particularly those determined by crystallography, only include information about part of the functional biological assembly. One thing to note is that the PDB can aid with this. Another note is many PDB entries are missing portions of the molecule that were not observed in the experiment. These include structures that include only alpha carbon positions, structures with missing loops, structures of individual domains, or subunits from a larger molecule. In addition, most of the crystallographic structure entries do not have information on hydrogen atoms [Goo].

4 Hadoop spark and pyspark

What is Hadoop

Hadoop is an open-source framework for writing and running distributed applications that process large amounts of data. Key aspects making it valuable such 1. Accessible 2.Robust 3. Scalable 4.simple [Lam10].

HDFS is used in haddop which is a filessystem and a MapReduce engine. With one master node and many worker nodes. The master node provides instructions to the worker nodes and computations are performed on the worker nodes. [HRJ17].

Mapper

Input key/value pairs are mapped to a set of key/value pairs. The mapper then sorts the key-value pairs by the keys. Partitioners are mainly responsible for providing intermediate key/values to the reducers [PBN12] [HRJ17].

Reducer

Firstly, the reducer combines data having the same key from different map functions. The values having the same key are reduced to a smaller set of values and output is produced [HRJ17].

What is Spark

Apache Spark is a popular open-source platform for large-scale data processing used for iterative machine learning tasks [MBY+16].

Spark is a cluster computing system providing APIs in Java, Scala, Python (pySpark), and R, along with an optimized engine that supports general execution graphs. Moreover, Spark is efficient at iterative computations so it is suited for the development of large-scale machine learning applications [MBY+16].

Spark is a quick and general engine used for analysing large-scale data stored across a cluster of computers. Spark uses in-memory cluster computing which is its most important feature for increasing the processing speed of an application. It combines SQL streaming and complex analytics [HRJ17].

Spark vs Hadoop

Hadoop Map Reduce	Spark
For Applications that repeatedly reuse the same set of data, map reduce is very inefficient.	Spark uses in-memory processing, reusing it for faster computation.
MapReduce is quite faster in batch processing.	As memory size is limited, it would be quite slower in batch processing of huge data set.
Data is stored in disk for processing.	Data is stored in main memory. As it is an inmemory computation engine entire data is copied.
Difficulty in processing and modifying data in real time due to its high latency.	Used to process and modify data in real time due to its low latency.
Predominantly used to process from bygone datasets.	Predominantly used for streaming, batch processing and machine learning
For fault tolerance, MapReduce uses replication.	For fault tolerance, Spark uses RDDs.
It merges and partitions shuffle files.	It does not merges and partition shuffle files.
Primarily disk based computation.	Primarily RAM based computation.

Table 5: Showing the differences between haddop and spark [HRJ17].

Number of words	Hadoop (Sec)	Spark(Sec)
100	79	28.841
1000	91	31.185
10000	96	35.181
100000	103	36.969
1000000	116	39.569

Table 6: Comparision of Execution time for wordcount program [HRJ17].

Number of words	Hadoop (Sec)	Spark(Sec)
5	2.541	0.9030
10	3.370	1.459
50	6.420	2.840
100	9.383	3.452
200	10.100	5.749

Table 7: Comparison of Execution time for logistic regression program [HRJ17].

Summarising the results shows Spark to be quicker in both experiments. Spark also provides an API for python which will be very helpful in this project seeing its easy nature to be able to read files and work with text-based files. Therefore I have decided to work with Pyspark for this project.

Software Architectural Bottlenecks

HDFS has scheduling delays in the architecture which results in cluster nodes waiting for new tasks as the access pattern is periodic.HDFS client code, serializes computation and I/O instead of decoupling and pipelining those operations. [SRC10].

Definition 4.1 (HDFS) *The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware*

Portability Limitations

Some performance-enhancing features in the filesystem are not available such as bypassing the filesystem page cache and transferring data directly from the disk into user buffers. Thus, HDFS implementation runs less efficiently and has higher processor usage than would otherwise be necessary [SRC10].

5 Software Development

5.1 Unit Tests

I used unittest — Unit testing framework python API to help me test my code. The point of unit testing is to isolate each section of the program and show that the individual section is functioning correctly. The way I based my unit test was around each function I have in my PoC program. I then thought of several cases that could happen and ensured that I wrote a test for them. For example, I created a test to test a method that retrieves the PDB files in a folder. more specifically this test checked various things for example it checked to see that it contained all the files with the extension .pdb from the correct folder. To test this I added a different extension file into that folder and ran the test looking at the length of the output to see if it is correct.

```
class TestGetAllPDBFiles(unittest.TestCase):
directory = ("/Users/vinaykakkar/Desktop/PROJECT-main/
ProofofConcepts/PDBontoaCluster/OriginalPDBs")

def test_checktypeofpdbs(self):
    test = Lines.getallpdbfiles(self.directory)
    lis = []
    self.assertTrue(type(test) is type(lis))

def test_checktypeofpdb(self):
    test = Lines.getallpdbfiles(self.directory)
    string = "string"
    self.assertTrue(type(test[0]) is type(string))

def test_checknumberofpdbs(self):
    test = Lines.getallpdbfiles(self.directory)
    self.assertEqual(len(test), 2)

def test_checkwrongdirectory(self):
    directory = ("Wrong")
    with self.assertRaises(Exception) as context:
        Lines.getallpdbfiles(directory)
```

Using these tests I made improvements to my code. For example, in the last test in the class TestGetAllPDBFiles, I check to see what happens if the directory provided is incorrect. At first, it errored with an ambiguous message which is not helpful. Once I ran the test I realized that I need to

create a check to see if the directory exists and if it doesn't then throw a readable and understandable exception. Yeilding in my program is more robust.

```
def getallpdbfiles(directory):
pdbfiles = []
## This line is checking to see if it exists first
## before trying to get all the .pdb files
if (Path(directory).is_dir()):
    for file in os.listdir(directory):
        filename = os.fsdecode(file)
        if filename.endswith(".pdb"):
            pdbfiles.append(filename)
else:
    raise Exception("No files found: check Directory")
return pdbfiles
```

5.2 Branching

The key role of branching is to allow more structure into the project which allows me to work on different aspects of a project at different times. As this project involves multiple parts for example when completing my tutorials on MapReduce I was already working on making a cluster for the first part of my PoC program. This gave me the freedom to easily work on different parts of the project very quickly. which allowed for a more structural progression compared to a linear block project progression such as completing the tutorial then completing the PoC the completing the reports in order.

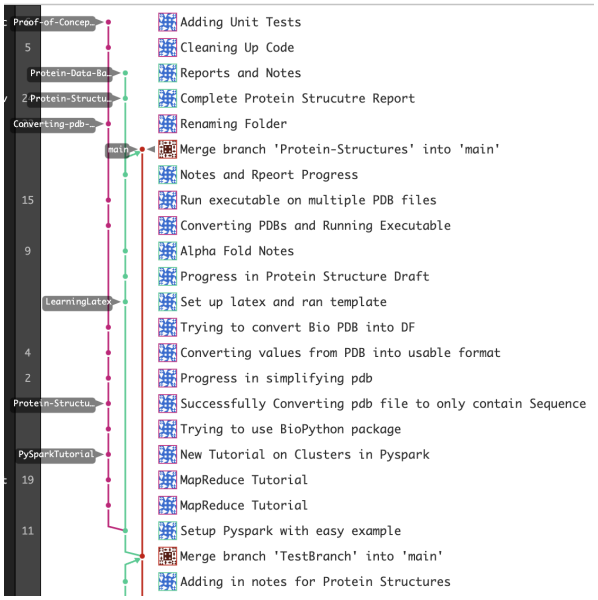


Figure 11: Showing some branching of my gitlab project.

The main two splits I have when working on my project a for the Poc and Reports/Research. Branching will help even more in the second term as there will be more features to add to the program

6 Planning and Time-Scale

My planning and Time-Scale will be based on the objectives I have yet to complete and any extensions I decided to include in my project.

6.1 Planning

The second term will be mostly working on the program with not much more to research on the biology aspects of the project. I will get a new diary that will contain my thoughts and ideas also noting down the struggles face. Summarising the objectives to complete and extensions:

- Provide some form of UI
- Perform/use user executables that are perfomred on pdb files which will yeild a result.
- Distribute clusters to multiple computers.
- Benchmarking.
- Mirror dataset from pdb site.
- Provide Guidelines for the executables.
- Improve UI
- A set of user executables that are ready and just need to be selected by the user.
- Implementation on a Public Mapreduce cluster e.g. Amazon.

I will also need to account for the final report and demo day presentation/poster at the end of the term.

looking back on Project Plan Timeline

Looking back to my project plan timeline/planning I was able to stick to most of all my suggested deadlines. However, writing reports have taken me much longer to complete than expected I underestimated how much time I will need to complete these reports and had to get an extension for my interim report. I will take this into count when creating a timeline for the second term

6.2 Time-Scale

The second term starts from 9th January till the Friday 24 March. This gives 12 weeks to complete these objectives and potentially extension.

When looking into the timeline 8, for weeks 1-2 I already have an incline on what API to use however I would like some more time to make sure I pick an easy approach. week 3-4 I want to be finished with the UI and also have user executables up and working that run on .pdb files. To do this I need to find user executables for .pdb files that are easy to set up. If I find that I can not do so I will seek help from my supervisor. Week 5-6 is now a good time to have benchmarks and tests ready for the main functional aspects of the project to be coming to an end. I can use these results to improve my program. Week 7-8 It is a good idea to start the report early as from prior experience writing reports takes longer than I expected so I am considering that. 9-12 is essentially working on my extension, guidelines, report, and poster.

7 Diary

I have a physical diary which i updated weekly, in this section i will blurt out my notes and explain the week at the end.

WeekOf:

26th September 2022

Questions asked in meeting with supervisor:

1. What does it mean to provide a framework
2. where can i read about functions of cellsand drug discoveries

Week	Description	Explanation
1-2	User Interface	Looking into how i want the UI to look and behave and what APIs to use. Once decided do tutorials for such APIs.
3-4	User Interface and Executables	Finish up the UI and set up a few executables which i allow the user to pick from.
5-6	Benchmarking	Setup and perform/document Benchmarking so that i can analyse performance depending on number of files.
7-8	Report	Start final report add in all privious work where need be and also include benchmarking.
9-10	Extension and guidlines	Implement one extension whilst writing up some guidlines on how to run/use the executable
11	Poster and Report	Create Poster for demo day whilst contiunuing work on report
12	Spark	Spare week

Table 8: The objectives to complete on a weekly basis

3. What is a user provided executable
4. Hadoop vs spark opinion
5. Explain differences of proof of concept programs

Next steps read chapters provided, look into alpha Fold, look into mahoot and start project plan. From this week i tried to soak in the core basis of my dissetation and planned to set things up such as gitlab and basic understanding of structural biology

3rd October 2022

Tried to make notes on large scale gene expressions but io am very confused. Set up gitlab, 2fa Auth, meeting next week, wont have draft ready so have questions ready to ask in meetin. I planed on getting my draft for my first report completed to show my supeviosur but was unable to do so i came up with questions to ask instead

10th October 2022

Questions asked in meeting with supervisor:

1. Confused about large scale expressions
2. Can i talk about other data type storage sites in my report about the PDB and file formats used
3. More clarity on the proof of concept programs

4. Where does alpha fold play in part

The summery of this week wes to start with earlier chapters from a book provided by the supivsor so that the later chapters will make sense. Try to get drafts ready to show the supervisor about the resarch i am doing first one being Protien structures.

17th October 2022

Notes: Keep everything on gitlab tht includes reports, notes, tests, tutorials, programs ect.. Remeber when writing report try to mention more relevant aspects in the department. Summery of the week Reports should be done by now but they are not so we need to ove on to the programs so that the whole project is not delayed.

24th October 2022

Questions asked in meeting with supervisor:

1. Do i need to look into experimental techniques used in labs to analyse protien structure
2. what are protein Folds
3. is it important to understand the chemical physical properties of amino acids

Notes need to set up pyspark and complete basic functionality so i get the gist of the api. At the same time need to keep reading on the material as it is starting to make sense.

31th October 2022

Notes spark set up and python set up, set up pyspark and completed tutorial that showed me setup and basic operations. Come up with questions for next weeks meeting with supervisor

7th November 2022

Questions asked in meeting with supervisor:

1. Do i need to read into peptide bonds
2. when is the fine line of when to stop looking into biology side of things
3. chapter 2 has a section about how shapes are formed do i need to read into this
4. is the difference between ahelix and bhelix somthing i should look into

Notes: Supervisor said they want to see my spark setup, read more into alpha fold. I only mention biology in my reports no need to go in depth to a degree. invite supervisor to gitlab. Summery: we need to focus more on spark and the program i have spent to much time researching and trying to learn the biology aspect side of the project.

14th November 2022

Notes: I have two user comits on gitlab need to see why this is the case. Clean up spark program to show supervisor next week. Coming to end of the term need to look into starting interim report

21th November 2022

Questions asked in meeting with supervisor:

1. Showed spark setup
2. clarify what is wrong with my proof of concpet programs

Notes: we dont need to convert the .pdb into a datafram that tries to split the file into many columns. We just need to split it line by line so the dataframe only has one column. Summery: Continue work on interim report as we are approaching the deadline

28th November 2022

Notes: completed Poc, working on interim report now will need to get extension to be able to finish on time

5th December 2022

Notes still working on Interim report

Bibliography

[AAB+19] Paul D. Adams, Pavel V. Afonine, Kumaran Baskaran, Helen M. Berman, John Berrisford, Gerard Bricogne, David G. Brown, Stephen K. Burley, Minyu Chen, Zukang Feng, Claus Flensburg, Aleksandras Gutmanas, Jeffrey C. Hoch, Yasuyo Ikegawa, Yumiko Kengaku, Eugene Krissinel, Genji Kurisu, Yuhe Liang, Dorothee Liebschner, Lora Mak, John L. Markley, Nigel W. Moriarty, Garib N. Murshudov, Martin Noble, Ezra Peisach, Irina Persikova, Billy K. Poon, Oleg V. Sobolev, Eldon L. Ulrich, Sameer Velankar, Clemens Vonrhein, John Westbrook, Marcin Wojdyr, Masashi Yokochi, and Jasmine Y. Young. Announcing mandatory submission of PDBx/mmCIF format files for crystallographic depositions to the Protein Data Bank (PDB). *Acta Crystallographica Section D Structural Biology*, 75(4):451–454, April 2019.

[ALFJ+17] Rebecca F. Alford, Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, P. Douglas Renfrew, Vikram K. Mulligan, Kalli Kappel, Jason W. Labonte, Michael S. Pacella, Richard Bonneau, Philip Bradley, Roland L. Jr. Dunbrack, Rhiju Das, David Baker, Brian Kuhlman, Tanja Kortemme, and Jeffrey J. Gray. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, June 2017. Publisher: American Chemical Society.

[BB21] Stephen K. Burley and Helen M. Berman. Open-access data: A cornerstone for artificial intelligence approaches to protein structure prediction. *Structure (London, England: 1993)*, 29(6):515–520, June 2021.

[BBB+21] Stephen K. Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V. Crichlow, Cole H. Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M. Duarte, Shuchismita Dutta, Zukang Feng, Sai Ganesan, David S. Goodsell, Sutapa Ghosh, Rachel Kramer Green, Vladimir Guranović, Dmytro Guzenko, Brian P. Hudson, Catherine L. Lawson, Yuhe Liang, Robert Lowe, Harry Namkoong, Ezra Peisach, Irina Persikova, Chris Randle, Alexander Rose, Yana Rose, Andrej Sali, Joan Segura, Monica Sekharan, Chenghua Shao, Yi-Ping Tao, Maria Voigt, John D. Westbrook, Jasmine Y. Young, Christine Zardecki, and Marina Zhuravleva. RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Research*, 49(D1):D437–D451, January 2021.

[BBB+22a] Stephen Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Henry Chao, Li Chen, Paul Craig, Gregg Crichlow, Kenneth Dalenberg, Jose Duarte, Shuchismita Dutta, Maryam Fayazi, Zukang Feng, Justin Flatt, Sai Ganesan, Sutapa Ghosh, David Goodsell, Rachel Kramer, Vladimir Guranovic, and Christine Zardecki. RCSB Protein Data Bank (RCSB.org): delivery of experimentally-determined PDB structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic Acids Research*, November 2022.

[BBB+22b] Stephen K. Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V. Crichlow, Jose M. Duarte, Shuchismita Dutta, Maryam Fayazi, Zukang Feng, Justin W. Flatt, Sai J. Ganesan, David S. Goodsell, Sutapa Ghosh, Rachel

Kramer Green, Vladimir Guranovic, Jeremy Henry, Brian P. Hudson, Catherine L. Lawson, Yuhe Liang, Robert Lowe, Ezra Peisach, Irina Persikova, Dennis W. Piehl, Yana Rose, Andrej Sali, Joan Segura, Monica Sekharan, Chenghua Shao, Brinda Vallat, Maria Voigt, John D. Westbrook, Shamara Whetstone, Jasmine Y. Young, and Christine Zardecki. RCSB Protein Data Bank: Celebrating 50 years of the PDB with new tools for understanding and visualizing biological macromolecules in 3D. *Protein Science*, 31(1):187–208, 2022. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pro.4213.

[BBR20] Sebastian Bittrich, Stephen K. Burley, and Alexander S. Rose. Real-time structural motif searching in proteins using an inverted index strategy. *PLOS Computational Biology*, 16(12):e1008502, December 2020. Publisher: Public Library of Science.

[BG21] Payam Behzadi and Márió Gajdács. Worldwide Protein Data Bank (wwPDB): A virtual treasure for research in biotechnology. *European Journal of Microbiology and Immunology*, 11(4):77–86, December 2021. Publisher: Akadémiai Kiadó Section: European Journal of Microbiology and Immunology.

[BKW+77] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The Protein Data Bank: a computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, 112(3):535–542, May 1977.

[BL22] Sarah E. Biehn and Steffen Lindert. Protein Structure Prediction with Mass Spectrometry Data. *Annual Review of Physical Chemistry*, 73(1):1–19, 2022. _eprint: https://doi.org/10.1146/annurev-physchem-082720-123928.

[Bou18] Boundless. 2.10: Atoms, Isotopes, Ions, and Molecules - Hydrogen Bonding and Van der Waals Forces, July 2018.

[BP03] Pierre Baldi and Gianluca Pollastri. The Principled Design of Large-Scale Recursive Neural Network Architectures–DAG-RNNs and the Protein Structure Prediction Problem. *NaN*, page 28, 2003.

[BPE22] Patrick Bryant, Gabriele Pozzati, and Arne Elofsson. Improved prediction of protein-protein interactions using AlphaFold2. *Nature Communications*, 13:1265, March 2022.

[BT98] Carl Ivar Branden and John Tooze. *Introduction to Protein Structure*. Garland Science, New York, 2 edition, December 1998.

[Bur21] Stephen K. Burley. Impact of structural biologists and the Protein Data Bank on small-molecule drug discovery and development. *The Journal of Biological Chemistry*, 296:100559, 2021.

[BWF+00] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, January 2000.

[DBB03] Cyril Dominguez, Rolf Boelens, and Alexandre M. J. J. Bonvin. HADDOCK: A Protein Protein Docking Approach Based on Biochemical or Biophysical Information. *Journal of the American Chemical Society*, 125(7):1731–1737, February 2003. Publisher: American Chemical Society.

[DITS22] Alessia David, Suhail Islam, Evgeny Tankhilevich, and Michael J. E. Sternberg. The AlphaFold Database of Protein Structures: A Biologist's Guide. *Journal of Molecular Biology*, 434(2):167336, January 2022.

[EWMR+06] Narayanan Eswar, Ben Webb, Marc A. Marti-Renom, M. S. Madhusudhan, David Eramian, Min-Yi Shen, Ursula Pieper, and Andrej Sali. Comparative protein structure modeling using Modeller. *Current Protocols in Bioinformatics*, Chapter 5:Unit–5.6, October 2006.

[Fel] Felix. A brief introduction to AlphaFold | Science | Felix Online.

[God22] W T. Godbey. Chapter 3 - Proteins. In W T. Godbey, editor, *Biotechnology and its Applications (Second Edition)*, pages 47–72. Academic Press, January 2022.

[Goo] David S. Goodsell. PDB101: Learn: Guide to Understanding PDB Data: Introduction.

[HRJ17] Akaash Vishal Hazarika, G Jagadeesh Sai Raghu Ram, and Eeti Jain. Performance comparision of Hadoop and spark engine. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pages 671–674, February 2017.

[JEP+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. Number: 7873 Publisher: Nature Publishing Group.

[KKN+06] Rimantas Kodzius, Miki Kojima, Hiromi Nishiyori, Mari Nakamura, Shiro Fukuda, Michihira Tagami, Daisuke Sasaki, Kengo Imamura, Chikatoshi Kai, Matthias Harbers, Yoshihide Hayashizaki, and Piero Carninci. CAGE: cap analysis of gene expression. *Nature Methods*, 3(3):211–222, March 2006. Number: 3 Publisher: Nature Publishing Group.

[KMY+15] Lawrence A. Kelley, Stefans Mezulis, Christopher M. Yates, Mark N. Wass, and Michael J. E. Sternberg. The Phyre2 web portal for protein modeling, prediction and analysis. *Nature Protocols*, 10(6):845–858, June 2015. Number: 6 Publisher: Nature Publishing Group.

[Lam10] Chuck Lam. *Hadoop in Action*. Simon and Schuster, November 2010. Google-Books-ID: 8DozEAAAQBAJ.

[LLV18] Patanachai Limpikirati, Tianying Liu, and Richard W. Vachet. Covalent labeling-mass spectrometry with non-specific reagents for studying protein structure and interactions. *Methods (San Diego, Calif.)*, 144:79–93, July 2018.

[LWL+20] Julia Koehler Leman, Brian D. Weitzner, Steven M. Lewis, Jared Adolf-Bryfogle, Nawsad Alam, Rebecca F. Alford, Melanie Aprahamian, David Baker, Kyle A. Barlow, Patrick Barth, Benjamin Basanta, Brian J. Bender, Kristin Blacklock, Jaume Bonet, Scott E. Boyken, Phil Bradley, Chris Bystroff, Patrick Conway, Seth Cooper, Bruno E. Correia, Brian Coventry, Rhiju Das, René M. De Jong, Frank DiMaio, Lorna Dsilva, Roland Dunbrack, Alexander S. Ford, Brandon Frenz, Darwin Y. Fu, Caleb Geniesse, Lukasz Goldschmidt, Ragul Gowthaman, Jeffrey J. Gray, Dominik Gront, Sharon Guffy, Scott Horowitz, Po-Ssu Huang, Thomas Huber, Tim M. Jacobs, Jeliazko R. Jeliazkov, David K. Johnson, Kalli Kappel, John Karanicolas, Hamed Khakzad, Karen R. Khar, Sagar D. Khare, Firas Khatib, Alisa Khramushin, Indigo C. King, Robert Kleffner, Brian Koepnick, Tanja Kortemme, Georg Kuenze, Brian Kuhlman, Daisuke Kuroda, Jason W. Labonte, Jason K. Lai, Gideon Lapidoth, Andrew Leaver-Fay, Steffen Lindert, Thomas Linsky, Nir London, Joseph H. Lubin, Sergey Lyskov, Jack Maguire, Lars Malmström, Enrique Marcos, Orly Marcu, Nicholas A. Marze, Jens Meiler, Rocco Moretti, Vikram Khipple Mulligan, Santrupti Nerli, Christoffer Norn, Shane Ó'Conchúir, Noah Ollikainen, Sergey Ovchinnikov, Michael S. Pacella, Xingjie Pan, Hahnbeom Park, Ryan E. Pavlovicz, Manasi Pethe, Brian G. Pierce, Kala Bharath Pilla, Barak Raveh, P. Douglas Renfrew, Shourya S. Roy Burman, Aliza Rubenstein, Marion F. Sauer, Andreas Scheck, William Schief, Ora Schueler-Furman, Yuval Sedan, Alexander M. Sevy, Nikolaos G. Sgourakis, Lei Shi, Justin B. Siegel, Daniel-Adriano Silva, Shannon Smith, Yifan Song, Amelie Stein, Maria Szegedy, Frank D. Teets, Summer B. Thyme, Ray Yu-Ruei Wang, Andrew Watkins, Lior Zimmerman, and Richard Bonneau. Macromolecular

modeling and design in Rosetta: recent methods and frameworks. *Nature Methods*, 17(7):665–680, July 2020. Number: 7 Publisher: Nature Publishing Group.

[LZD+20] Joseph H. Lubin, Christine Zardecki, Elliott M. Dolan, Changpeng Lu, Zhuofan Shen, Shuchismita Dutta, John D. Westbrook, Brian P. Hudson, David S. Goodsell, Jonathan K. Williams, Maria Voigt, Vidur Sarma, Lingjun Xie, Thejasvi Venkatachalam, Steven Arnold, Luz Helena Alfaro Alvarado, Kevin Catalfano, Aaliyah Khan, Erika McCarthy, Sophia Staggers, Brea Tinsley, Alan Trudeau, Jitendra Singh, Lindsey Whitmore, Helen Zheng, Matthew Benedek, Jenna Currier, Mark Dresel, Ashish Duvvuru, Britney Dyszel, Emily Fingar, Elizabeth M. Hennen, Michael Kirsch, Ali A. Khan, Charlotte Labrie-Cleary, Stephanie Laporte, Evan Lenkeit, Kailey Martin, Marilyn Orellana, Melanie Ortiz-Alvarez de la Campa, Isaac Paredes, Baleigh Wheeler, Allison Rupert, Andrew Sam, Katherine See, Santiago Soto Zapata, Paul A. Craig, Bonnie L. Hall, Jennifer Jiang, Julia R. Koeppe, Stephen A. Mills, Michael J. Pikaart, Rebecca Roberts, Yana Bromberg, J. Steen Hoyer, Siobain Duffy, Jay Tischfield, Francesc X. Ruiz, Eddy Arnold, Jean Baum, Jesse Sandberg, Grace Brannigan, Sagar D. Khare, and Stephen K. Burley. Evolution of the SARS-CoV-2 proteome in three dimensions (3D) during the first six months of the COVID-19 pandemic. *bioRxiv: The Preprint Server for Biology*, page 2020.12.01.406637, December 2020.

[LZG20] Xiaoran Roger Liu, Mengru Mira Zhang, and Michael L. Gross. Mass Spectrometry-Based Protein Footprinting for Higher-Order Structure Analysis: Fundamentals and Applications. *Chemical Reviews*, 120(10):4355–4454, May 2020. Publisher: American Chemical Society.

[MBY+16] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, D. B. Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, and Ameet Talwalkar. MLlib: Machine Learning in Apache Spark. *Journal of Machine Learning Research*, 17(34):1–7, 2016.

[NZLJ22] Ruth Nussinov, Mingzhen Zhang, Yonglan Liu, and Hyunbum Jang. AlphaFold, Artificial Intelligence (AI), and Allostery. *The Journal of Physical Chemistry B*, 126(34):6372–6383, September 2022. Publisher: American Chemical Society.

[OM06] Michał J. Okoniewski and Crispin J. Miller. Hybridization interactions between probesets in short oligo microarrays lead to spurious correlations. *BMC Bioinformatics*, 7(1):276, June 2006.

[OR15] Robert J. Ouellette and J. David Rawn. 14 - Amino Acids, Peptides, and Proteins. In Robert J. Ouellette and J. David Rawn, editors, *Principles of Organic Chemistry*, pages 371–396. Elsevier, Boston, January 2015.

[PBN12] Aditya B. Patel, Manashvi Birla, and Ushma Nair. Addressing big data problem using Hadoop and Map Reduce. In *2012 Nirma University International Conference on Engineering (NUiCONE)*, pages 1–5, December 2012. ISSN: 2375-1282.

[RBL+02] Jeannette Reinartz, Eddy Bruyns, Jing-Zhong Lin, Tim Burcham, Sydney Brenner, Ben Bowen, Michael Kramer, and Rick Woychik. Massively parallel signature sequencing (MPSS) as a tool for in-depth quantitative gene expression profiling in all organisms. *Briefings in Functional Genomics*, 1(1):95–104, February 2002.

[RDL+21] Yana Rose, Jose M. Duarte, Robert Lowe, Joan Segura, Chunxiao Bi, Charmi Bhikadiya, Li Chen, Alexander S. Rose, Sebastian Bittrich, Stephen K. Burley, and John D. Westbrook. RCSB Protein Data Bank: Architectural Advances Towards Integrated Searching and Efficient Access to Macromolecular Structure Data from the PDB Archive. *Journal of Molecular Biology*, 433(11):166704, May 2021.

[RLW+12] Daniel Russel, Keren Lasker, Ben Webb, Javier Velázquez-Muriel, Elina Tjioe, Dina Schneidman-Duhovny, Bret Peterson, and Andrej Sali. Putting the Pieces Together:

Integrative Modeling Platform Software for Structure Determination of Macromolecular Assemblies. *PLOS Biology*, 10(1):e1001244, January 2012. Publisher: Public Library of Science.

[RRG07] Thomas E. Royce, Joel S. Rozowsky, and Mark B. Gerstein. Toward a universal microarray: prediction of gene expression through nearest-neighbor probe sequence identification. *Nucleic Acids Research*, 35(15):e99, 2007.

[SC10] Harry W Schroeder and Lisa Cavacini. Structure and Function of Immunoglobulins. *The Journal of allergy and clinical immunology*, 125(2 0 2):S41–S52, February 2010.

[SFB04] Peter D. Sun, Christine E. Foster, and Jeffrey C. Boyington. Overview of Protein Structural and Functional Folds. *Current Protocols in Protein Science*, 35(1):1711–171189, February 2004.

[SL20] Justin T. Seffernick and Steffen Lindert. Hybrid methods for combined experimental and computational determination of protein structure. *The Journal of Chemical Physics*, 153(24):240901, December 2020. Publisher: American Institute of Physics.

[SRC10] Jeffrey Shafer, Scott Rixner, and Alan L. Cox. The Hadoop distributed filesystem: Balancing portability and performance. In *2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS)*, pages 122–133, March 2010.

[TAW+21] Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, Sameer Velankar, Gerard J. Kleywegt, Alex Bateman, Richard Evans, Alexander Pritzel, Michael Figurnov, Olaf Ronneberger, Russ Bates, Simon A. A. Kohl, Anna Potapenko, Andrew J. Ballard, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Ellen Clancy, David Reiman, Stig Petersen, Andrew W. Senior, Koray Kavukcuoglu, Ewan Birney, Pushmeet Kohli, John Jumper, and Demis Hassabis. Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596, August 2021. Number: 7873 Publisher: Nature Publishing Group.

[vdABH17] Wil M. P. van der Aalst, Martin Bichler, and Armin Heinzl. Responsible Data Science. *Business & Information Systems Engineering*, 59(5):311–313, October 2017.

[WDA+16] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, March 2016. Number: 1 Publisher: Nature Publishing Group.

[WF03] John D. Westbrook and Paula M. D. Fitzgerald. The PDB Format, mmCIF Formats, and Other Data Formats. In *Structural Bioinformatics*, pages 159–179. John Wiley & Sons, Ltd, 2003. Section: 8 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0471721204.ch8.

[WGS09] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, January 2009. Number: 1 Publisher: Nature Publishing Group.

[WIN+05] John Westbrook, Nobutoshi Ito, Haruki Nakamura, Kim Henrick, and Helen M. Berman. PDBML: the representation of archival macromolecular structure data in XML. *Bioinformatics*, 21(7):988–992, April 2005.

[WSHB20] John D. Westbrook, Rose Soskind, Brian P. Hudson, and Stephen K. Burley. Impact of the Protein Data Bank on antineoplastic approvals. *Drug Discovery Today*, 25(5):837–850, May 2020.

[YYR+15] Jianyi Yang, Renxiang Yan, Ambrish Roy, Dong Xu, Jonathan Poisson, and Yang Zhang. The I-TASSER Suite: protein structure and function prediction. *Nature Methods*, 12(1):7–8, January 2015. Number: 1 Publisher: Nature Publishing Group.

[Zve08] Marketa J. Zvelebil. *Understanding bioinformatics / Marketa Zvelebil & Jeremy O. Baum.* Garland Science/Taylor & Francis Group, Garland Science, Taylor & Francis Group, New York, 2008.