



MDMC

Master in Data Management
and Curation

MASTER IN DATA MANAGEMENT AND
CURATION

**Data Extraction Tool for Protein
Crystallization Conditions**

Supervisor(s):

Dr. Andrea DALLE VEDOVE,

Dr. Valerio PIOMPONI

Candidate:

Ruth NANA NJANTANG EPSE OSAZUWA

2025–2026





Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA

This course has been funded by the European Union – NextGenerationEU within the projects:

- *NFFA-DI cod. IR00000015 - Missione 4, “Istruzione e Ricerca” – Componente 2, “Dalla ricerca all’impresa” – Linea di investimento 3.1, “Fondo per la realizzazione di un sistema integrato di infrastrutture di ricerca e innovazione” – Azione 3.1.1, “Creazione di nuove IR o potenziamento di quelle esistenti che concorrono agli obiettivi di Eccellenza Scientifica di Horizon Europe e costituzione di reti” (CUP B53C22004310006)*

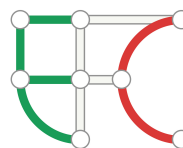
[NFFA-DI : Nano Foundries Fine Analysis - Digital Infrastructure](#)

- *EFC cod. SSU2024 – 00002 - Missione 4, “Istruzione e Ricerca” – Componente 1, “Potenziamento dell’offerta dei servizi all’istruzione: dagli asili nido all’ università” – Linea di investimento 3.4, “Didattica e competenze universitarie avanzate” – Sub-Investimento, “Rafforzamento delle scuole universitarie superiori” (CUP G97G24000100007)*

[EFC : Educating Future Citizen](#)



nffa-di



Author's Declaration

I, **NANA NJANTANG EPSE OSAZUWA Ruth**, declare that this thesis entitled, 'Data Extraction Tool for Protein Crystallization Conditions' and the work presented therein are my own.

I certify that:

- This work was performed entirely or principally during the MDMC internship in the **Laboratory of Data Engineering (LADE) at Area Science Park**.
- If any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have relied on the published work of others, this is always clearly attributed.
- Where the thesis is based on work that I have done in collaboration with others, I have made clear exactly what was done by others and what I contributed.
- With respect to AI technologies:
 - I acknowledge the use of claude.ai (<https://claude.ai>) and OpenAI's ChatGPT (<https://chat.openai.com>) to identify improvements in the writing style.
 - I acknowledge the use of OpenAI's ChatGPT (<https://chat.openai.com>) and notebook LM (<https://notebooklm.google.com>) as a source of information to create materials that will be used in my own words.
- Where I have quoted from the work of others, the source is always cited. Except for such quotations, this thesis is entirely my own work.
- I have acknowledged all major sources of help.



Signed

Date: July 1, 2026

“Nothing in life is to be feared, it is only to be understood. Now is the time to understand more so that we may fear less.”

MARIE CURIE

Acknowledgments

I would like to express my deepest gratitude to my supervisors, **Dr. Valerio Piomponi** and **Dr. Andrea Dalle Vedove**, for their guidance, support, and scientific insight throughout this project. Their expertise and dedication have been instrumental in shaping both the direction and the quality of this work.

I warmly thank the program coordinator, **Dr. Mariarita de Luca**, and all the lecturers and tutors of the MDMC for the knowledge shared and for the continued support throughout the program.

I thank the Laboratory of Data Engineering (LADE) at RIT, Area Science Park, for sponsorship and for providing all logistical resources that contributed to the successful completion of this project.

I also would like to express my appreciation to my friends and colleagues for their constant encouragement and stimulating exchanges that made this journey exciting and more rewarding.

Finally, and most profoundly, I wish to thank my beloved husband for his patience, support, and understanding throughout this journey. His presence has been my greatest source of strength.

Abstract

Despite recent progress in identifying the factors relevant to protein crystallization, the successful prediction of suitable crystallization conditions remains a significant challenge in structural biology. In this work, we present an automated data mining pipeline, developed with Python 3, that leverages the wealth of experimental metadata archived in the Protein Data Bank to improve crystallization success rates through the homology guided approach.

Given a query sequence, the pipeline performs a sequence similarity search against all X-ray diffraction determined structures in the PDB, retrieves and standardizes the associated crystallization metadata including polymer type, assembly, pH, temperature, crystallization method, list of compounds and their concentrations, etc. and compiles the results into a structured, annotated dataset.

The extracted data are filtered, merged, and presented through three complementary outputs: a CSV file, a colored PDF table summarizing all unique crystallization conditions alongside key biological and structural metadata, and scatter plots visualizing the distribution of pH against PEG concentration, colored by sequence similarity score to highlight the most relevant conditions.

To maximize accessibility, the complete pipeline was deployed as an interactive web application built on the Django framework, allowing users to visualize results and directly download all generated output files. By integrating crystallization data from the PDB archive into a FAIR compliant resource (downloadable PDF summary reports and an interactive analytical plotting tool) the platform streamlines the design of new crystallization trials. It enables researchers to skip initial trial-and-error stages, supports the rational design of crystallization experiments, and contributes to a deeper understanding of the physicochemical principles governing macromolecular crystal growth.

Contents

1	General Introduction	1
1.1	Background	1
1.2	Problem Statement	1
1.3	Objective	2
1.4	Thesis organization (Structure)	2
2	Fundamentals of macromolecular Crystallization	4
2.1	Global Approach of Crystallography	4
2.1.1	Macromolecules Structure Pipeline	4
2.1.2	Crystallization of macromolecules	5
2.1.3	Optimization of crystallization conditions for biological macromolecules	10
2.1.4	Main Factors Influencing Macromolecular Crystallization	11
2.2	The Protein Data Bank(PDB)	13
2.2.1	The Metadata and the Experimental data in the PDB	13
2.2.2	The PDB Application Programming Interfaces (APIs)	14
3	Extraction of Macromolecules Crystallization Data Method	17
3.1	General Pipeline from Macromolecule to Crystallization Cocktail	17
3.2	The Sequence Homology Search	18
3.3	Extraction of Structures Metadata from the PDB	20
3.3.1	Input Processing and Sequence Search	22
3.3.2	Optimization of Data Retrieval and Caching	22
3.3.3	Crystallization Parameters Extraction and Text Mining	23
3.3.4	Data Refinement and Output	24
3.4	Standardization of Crystallization details	25
3.4.1	Parsing Crystallization Details into a List of Compounds	25
3.4.2	Enrichment of Crystallization Details with the Standardized list of Compound	26
3.5	Visualization and Reporting	27
3.5.1	Data Preprocessing	27
3.5.2	High Resolution Output and Reporting	28
4	Results and Discussions	30
4.1	PDB Extraction with Sequence Alignment Score	30
4.2	Extrated Metadata for Crystallization Optimization	31
4.2.1	Pubmed_id	32
4.2.2	Polymer type	32

4.2.3	Assembly	33
4.2.4	Ligands	33
4.2.5	Method	33
4.2.6	Crystallization Cocktail	33
4.3	Data Analysis	34
4.4	Crystallization Cocktail Webpage	37
4.5	FAIR Principles	39
4.5.1	Implementation of FAIR for Data Management	40
4.5.2	FAIR for Research Software Assessment: PCDE Project	43
5	Conclusion	46
A	Multiple Sequence Fasta file Scripts	48
A.1	scripts_fasta_file Folder	48
A.1.1	main.py	48
A.1.2	main2.py	49

Chapter 1

General Introduction

1.1 Background

Crystallization is usually the bottleneck process in the determination of the three dimensional structure of a macromolecule through X-ray diffraction (XRD). The field encompasses not only proteins but also nucleic acids and large assemblies such as viruses and ribosomes, and draws on a broad mix of principles, empirical knowledge, and practical intuition.

According to Kwon et al. 2013, one of the main difficulties in macromolecular crystallization is the setting up cocktails that produce a single large crystal for collection [1]. No comprehensive theoretical framework currently guides the process, and the fundamental data needed to build one is still being accumulated. As a result, crystal growth is largely trial-and-error, requiring patience and persistence from the researcher, in addition to the large amount of materials. At its core, successful crystallization depends on bringing the macromolecule to a supersaturated state without disrupting its native conformation.

This is achieved by introducing mild precipitating agents and by carefully adjusting physical parameters, including temperature, ionic strength, and pH. Beyond these primary variables, the structural integrity of the macromolecule can be further influenced by the presence of metal ions, inhibitors, co-factors, or other small molecules, all of which must be considered when designing a crystallization strategy.

1.2 Problem Statement

Macromolecular crystallization is commonly achieved by carrying out an extensive series, or establishing a vast matrix, of crystallization trials, evaluating the results, and using the information obtained to improve conditions in successive rounds of trials (conditions that ultimately yield the best crystals for the collection of diffraction data) [2]. Because the number of variables is so large and because the ranges are so broad, experience and insight in designing and evaluating individual and collective trials become an important consideration [3].

A key factor in unearthing those principles is gathering enough information to tease out the complicated interactions between the crystallization parameter space and crystal formation. The Protein Data Bank (PDB) is an incredibly rich source of data on successful crystallization parameters, as it contains information on crystallization conditions in a free text field. Mining this free text field to get information about crystallization cocktails can provide insight about the conditions in which macromolecular diffraction quality crystals have formed. However, it is difficult to analyze these details across the entire PDB dataset. Firstly, because of the lack of standardized reporting within the free text field. In addition, there are inconsistencies in compound identity naming conventions (including typos and misspellings), in punctuation and spacing, and in exactly what information is listed [4].

There has been a longstanding interest in developing ways to better understand which parameters in the crystallization space enable macromolecular crystallization, but it remains a fundamental challenge to predict crystallization conditions that will be successful given details of the macromolecular target. In this context, computational approaches offer an opportunity to systematically retrieve and organize crystallization data. By combining sequence-based homology searches with automated access to public databases, it becomes possible to aggregate information from structurally related proteins and extract experimentally validated conditions. Such an approach can help transform fragmented data into a coherent and machine-readable resource, enabling more informed decision-making in experimental design.

1.3 Objective

This work describes the development of a computational tool that automatically retrieves, organizes, and analyzes data related to the crystallization conditions of macromolecules from the PDB repository. The specific objectives are:

- Retrieve all PDB entries with sequence similar to the query sequence with the structures based XRD experiment.
- Collect structural metadata and experimental parameters, with an emphasis on crystallization conditions (pH, temperature, precipitants, buffers, additives, etc.).
- Organize extracted information into machine readable formats (e.g. csv, pdf, etc.), with a standardized terminology and units across datasets.

1.4 Thesis organization (Structure)

The thesis opens with the general introduction in chapter 1 which presents the problem statement and objectives. Chapter 2 provides a comprehensive overview of macromolecular crystallization, covering the physicochemical principles that govern crystal nucleation and growth, the main experimental approaches used to screen and optimize crystallization conditions, and the role of the Protein Data Bank as the primary repository of successful crystallization metadata.

Chapter 3 presents the technical core of the work, detailing the automated pipeline

developed to address the crystallization optimization problem. It describes the complete workflow from sequence input to data output.

Chapter 4 presents the output files generated by the pipeline and demonstrates how the extracted information can be interpreted and applied for the optimization of crystallization conditions. the thesis round up with a conclusion which is chapter 5.

Chapter 2

Fundamentals of macromolecular Crystallization

Understanding the three dimensional structure of proteins is essential for elucidating their biological function and for applications ranging from drug design to enzyme engineering. Among the available structural determination techniques, X-ray diffraction (XRD) remains one of the most widely used methods. However, it is critically dependent on the successful crystallization of the target macromolecule. This chapter introduces the theoretical background underlying the development of a data extraction tool for macromolecule crystallization conditions. It reviews the principles of crystallization, the organization and content of structural databases, and the computational strategies used for sequence comparison, data retrieval, and standardization.

2.1 Global Approach of Crystallography

2.1.1 Macromolecules Structure Pipeline

Throughout the years, structural determination of biological macromolecules and their macromolecular complexes has been developed primarily by XRD being the most widely used. With the increasing availability of synchrotron radiation facilities, X-ray analysis remains the most powerful method to obtain high resolution 3D structures of macromolecules [5].

In the process of obtaining a crystal structure, the crucial thing you need is a crystal. A key bottleneck of crystallographic methods is finding conditions in which a sample will crystallize. Figure 2.1 below briefly highlights the steps of the macromolecular structure determination pipeline.



Figure 2.1: *Experimental pipeline to produce structural information of macromolecules based on XRD method* [6]

Starting from a purified Macromolecule solution, the sample is subjected to extensive crystallization screening using a wide range of chemical conditions to identify those that promote crystal formation. Once suitable conditions are found, the protein molecules assemble into an ordered crystal lattice.

The crystal is then exposed to an X-ray beam, producing a diffraction pattern as the X-rays scatter off the electrons in the crystal. The resulting diffraction data is processed to calculate an electron density map using mathematical transformations. Subsequently, a structural model of the macromolecule is built by fitting amino acid residues to the electron density map.

Finally, the model undergoes iterative refinement and validation to optimize its agreement with the experimental data, yielding the final three dimensional structure of the protein at atomic resolution.

2.1.2 Crystallization of macromolecules

Macromolecular crystallization including the crystallization of proteins, nucleic acids, and larger macromolecular assemblies such as viruses and ribosomes. Crystallization of biomolecules requires a balance between stabilizing and solubilizing the sample coupled with driving toward an ordered aggregate, resulting in a lattice held together by a periodic network of sparse and weak intermolecular interactions [7]. For biomolecules, crystallization often proceeds through extensive experimentation, but there are guiding principles that can be employed to increase the chance of success. There is an extensive history of literature that describes methods to crystallize biomolecules[8, 9, 10].

Crystallization is based on a diverse set of principles, experiences, and methodologies. The main principles governing macromolecular crystallization are:

- Searching, as systematically as possible, the ranges of the individual parameters that influence crystal formation,
- Finding a set or multiple sets of factors that yield some kind of crystals,
- Optimizing individual variables to obtain the best possible crystals.

A detailed study of the crystallization process provides a better understanding of the unique features of proteins, nucleic acids, and macromolecular assemblies that influence

their ability to crystallize. The following steps are particularly important in addressing the crystallization problem [3]:

- **Homogeneity:** begin with as pure and uniform a population of a molecular species as possible;
- **Solubility:** dissolve the macromolecule to a high concentration without the formation of aggregates, precipitate, or other phases. The methods used to investigate the purity of samples used in structural studies have been discussed [11];
- **Stability:** do whatever is necessary to maintain the macromolecules as stable and unchanging as possible;
- **Supersaturation:** alter the properties of the solution to obtain a system which is appropriately supersaturated with respect to the macromolecule;
- **Association:** try to promote the orderly association of the macromolecules while avoiding precipitate, nonspecific aggregation, or phase separation;
- **Nucleation** tries to promote the formation of a few critical nuclei in a controlled manner;
- **Variety:** explore as many possibilities and opportunities as possible in terms of biochemical, chemical and physical parameters;
- **Control:** maintain the system at an optimal state, without fluctuations or perturbations, during the course of crystallization;
- **Impurity:** discourage the presence of impurities in the mother liquor and the incorporation of impurities and foreign materials into the lattice;
- **Preservation:** once the crystals have grown, they are protected from shock and disruption, and their stability is maintained.

In practice, it is the responsibility of the individual investigator to find practical means to apply these principles, depending on a specific problem which is of critical importance and which will have the greatest influence on the likelihood of crystallization success. Figure 2.2 gives a global view of the steps that produce the successful crystallization of macromolecules.

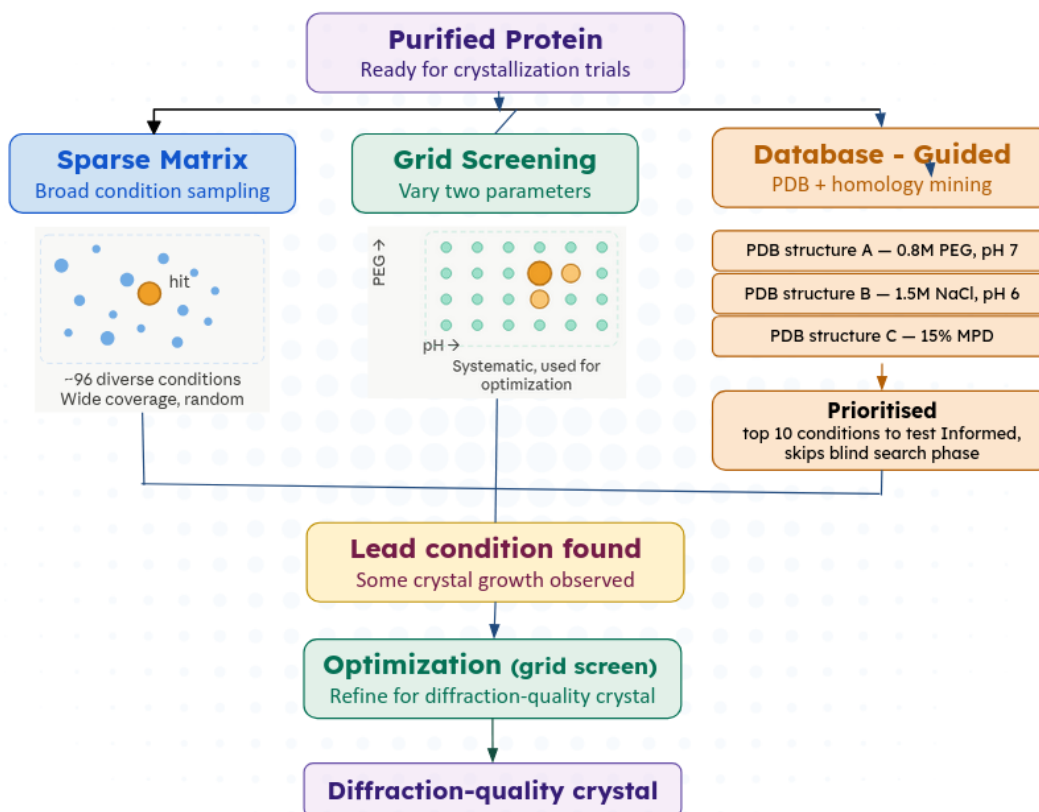


Figure 2.2: Three main experimental approaches that can be used to optimized crystallization conditions

The main approaches taken into account in the pursuit of protein crystals for an X-ray diffraction investigation are:

a. Purification and characterization of proteins

Crystallization, including protein crystallization, is a means of purifying a chemical compound as it demonstrate that a compound had been obtained in a completely pure form. Purification and characterization are critical steps before protein crystallization because crystal quality strongly depends on protein purity, homogeneity, stability, and structural integrity. In general, the likelihood of success in crystal growth is greatly advanced by increased homogeneity of the sample. Investment in further purification is always warranted, and usually profitable when every effort to crystallize a macromolecule fails [3].

When all approaches to crystallization have been exhausted, then there are two remaining options: further purifying the protein or modifying the protein. The latter may be accomplished by genetic means if the protein has been created by recombinant DNA techniques [12].

Another strategy that has been used for recalcitrant proteins is to combine them in some manner with a second protein so that the complex of the two provides an additional chance for success[13]. This is particularly the case of membrane proteins. In these cases the antibody fragment enhanced the solubility

of the otherwise insoluble protein and provided additional lattice contacts in the resultant crystals. Histidine tags and proteins conjugated with a second protein, such as maltose binding protein, are frequently produced as a basis for purification procedures for recombinant proteins. Through truncations, mutations, chimeric conjugates and many other protein engineering contrivances, the probability of crystallization may be significantly enhanced [14].

The solubility of proteins in water depends on properties such as temperature, pH, and the presence of other solution components as well as amino acid composition. When the concentration of a protein solution is brought above its solubility limit, the solution becomes supersaturated. At this point, it begins to aggregate. Aggregation occurs in two stages as a nucleation and as a growth.

During nucleation, the molecules associate into a stable complex as an amorphous precipitate or as a microcrystals. Amorphous precipitates tend to predominate when the protein concentration is well above saturation. In addition, crystals grow much slower than amorphous precipitates do, so when the concentration of a protein is brought above its saturation point too quickly, precipitation will again predominate. In the metastable region, if a few nuclei are present, they will continue to grow, but without spontaneous formation of new nuclei.

As crystals grow the concentration of protein in solution decreases and supersaturation will fall. Growth will be slow and orderly and will produce the fewest and largest single crystals. Thus, the most general crystallization strategy is to bring the protein to the point only slightly above its saturation point as slowly as possible. Figure 2.3 shows details of the proteins crystallization processes.

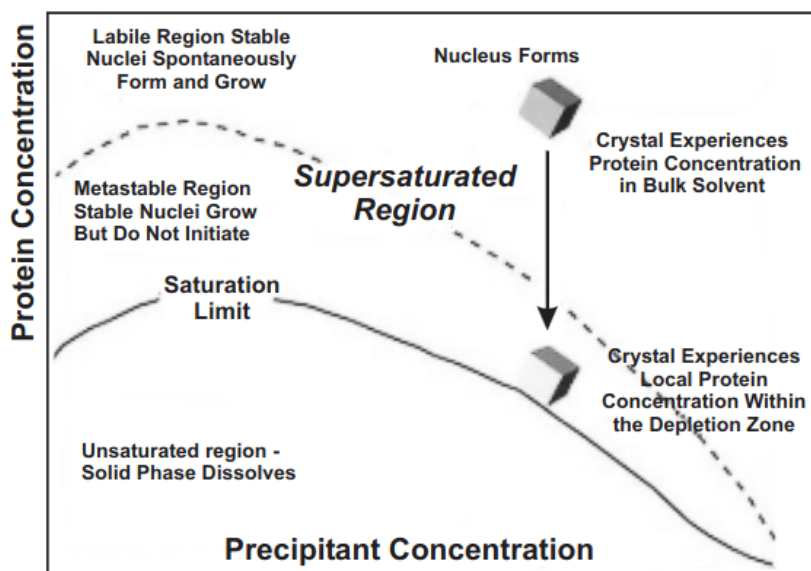


Figure 2.3: Phase diagram of protein crystallization showing the relationship between protein concentration and precipitant concentration [3]

- b. Test screening of the macromolecules solubility considering the precipitants agents and other solution components

Usually this step consist in the identification of chemical, biochemical and physical conditions that yield some crystalline material, although that may be entirely inadequate. There are three main experimental approaches used to screen for crystallization conditions (see figure 2.3), each with a different philosophy about how to explore the vast parameter space.

- Sparse matrix screening is the most common starting point. Rather than systematically varying every parameter, it samples a wide diversity of conditions different buffers, salts, precipitants, and pH values chosen based on conditions that have historically worked for other proteins.
- a shotgun approach: Currently, a wide variety of crystallization screening kits from numerous companies are available on the commercial market. The availability and ease of use of these relatively modestly priced kits, which may be used in conjunction with a variety of crystallization methods (hanging and sitting drop vapor diffusion, dialysis etc.) make them the first tool of choice in attacking a new crystallization problem. With these kits, nothing more is required than combining a series of potential crystallization solutions with one's protein of interest using a micropipette, sealing the samples and waiting for success.
- Computational and database guided screening, which connects directly to the BLAST and PDB workflow. By mining successful crystallization conditions from homologous structures in the PDB, you can prioritize which conditions to test first, effectively skipping the blind sparse matrix phase and starting closer to a working condition.

c. Optimization

Optimization is commonly taken to mean adjusting the parameters of crystallization conditions initially obtained from screening matrices with the objective of discovering improved conditions that ultimately yield the best crystals for diffraction data collection.

The initial crystals obtained from the screens are generally of insufficient quality. Optimization is in a sense refinement, but it is complicated somewhat because the parameters are almost certainly interdependent. McPherson and Cudney in 2014 have made an attempt to provide guidance on how optimization might best be applied to crystal-growth problems, and what parameters and factors might most necessarily be explored to accelerate and achieve success [15].

In 2016, Dinc et al proposed an experimental design method called “Associative Experimental Design (AED)” and an optimization method that includes eliminating prohibited combinations and prioritizing reagents based on AED analysis of results from protein crystallization experiments. AED generates candidate cocktails based on the initial screening results. These results are analyzed to determine those screening factors in chemical space that are most likely leading to higher scoring outcomes, crystals [16].

2.1.3 Optimization of crystallization conditions for biological macromolecules

Optimization, as it is often practiced, is illustrated schematically in figure 1.4 below and is in principle relatively straightforward. The parameters defining the initial conditions are identified first. (pH, precipitant type, precipitant concentration, temperature, ion concentration, etc.)

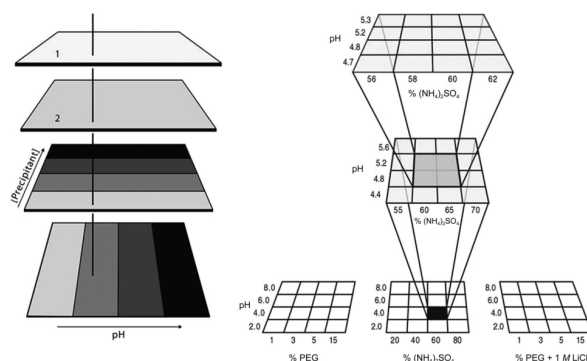


Figure 2.4: *Schematic illustration of the successive grid search strategy for protein crystallization redrawn from Cox and Weber in 1988 [17]*

On the left of the figure, components of the grid search are displayed separately. The bottom square shows the variation in pH across the columns. The square above shows the variation in precipitant concentration in the rows. The combination of these two layers produces the pH versus precipitant grid that serves as the basis for the two-dimensional crystallization approach. Fixed concentrations of other reagents can be added unto this grid as indicated by the upper squares labeled 1 and 2. The diagram on the right illustrates how solution parameters are chosen using this strategy. Broad screen experiments (shown at the bottom) are setup using three different precipitating agents. Tight ranges of pH and precipitant concentration are centered about the conditions in the droplet crystals.

Optimization requires the composition of a vast number of solutions that must be formulated or purchased, and the use of robotics in optimization presents as many problems as it solves, at least at this point in time. Making up a myriad of solutions, adjusting their pH to exact values and so on is tedious. Optimization can be achieved by comparing the conditions of all of the successful trials and look for common characteristics.

If all of the successes used a polymer such as polyethylene glycol (PEG) as the precipitant and no crystals were grown from salt solutions, then clearly the focus must be on PEG. If crystals were grown only in narrow ranges about, say, pH 5 or pH 8.2, then it is reasonable to assume that the crystals at the two pH values are likely to be fundamentally different. Initially, at least, optimization should be carried out around both of those two pH values.

If a significant number of initial successes, for example, included Mg^{2+} or Ca^{2+} or some other ions, then the response would be straightforward. Examining the various

conditions and looking for commonalities may allow a large set of possibilities to be reduced to a manageable set.

2.1.4 Main Factors Influencing Macromolecular Crystallization

There are many factors that affect the crystallization of macromolecules or the probability of its occurring at all, the nucleation probability and rate, the crystal-growth rate, and/or the ultimate sizes and quality of the products. The concentration of the macromolecule, the pH and salt, or the concentrations of other precipitants, are of great importance. Other parameters may be less important in general, but can play crucial roles in specific cases. The presence or absence of cofactors, ligands or inhibitors, the variety of salts or buffers, the equilibration technique used, the temperature and the presence of detergents are all pertinent considerations. Parameters of somewhat lesser significance include gravity, electric and magnetic fields, and viscosity. It can, in general, not be predicted which of these many variables may be of importance for a particular macromolecule, and the suspected influence of any one must, in general, be evaluated by empirical trials. The most probable effectors are those which maintain the protein in a single, homogeneous and invariant state. Certain chemical compounds or small molecules may have dramatic effects on the success with which individual proteins crystallize. Additives, as they are often called, can be decisive in macromolecular crystallization [18, 19]. Numerous cases have, however, been reported in which small, and sometimes large, molecules were observed to make crucial interactions between macromolecules in the crystal that either helped to guide or secure formation of the lattice [20]. They simply provided essential or at least helpful cross-links within the crystal. Additives that are used in crystallization may be classified into eight categories [3].

a. Physiologically or biochemically relevant small molecules

They bind either to enzyme active sites or to defined regions on macromolecules, stabilizing more uniform conformations or shifting the structure into alternative states. As a result, the protein–ligand complex often becomes more monodisperse and less dynamic.

The relevant ligands are protein-specific, and their inclusion in mother liquors can be guided by rational analysis grounded in the protein's enzymology and biochemistry. In practice, this means considering the full range of potential ligands and incorporating them into the crystallization screening conditions. The concentrations of protein ligands (inhibitors, coenzymes, substrates etc.) or metal ions (generally divalent, Mg^{2+} , Ca^{2+} , Mn^{2+} etc.) in the mother liquor do not generally require extensive optimization.

Once it has been determined that they are required or useful, then it can be assumed that there exists a specific association constant that defines their affinity. It is essential that the concentration of the ions, ligands or cofactors exceed the concentrations necessary to saturate the binding sites of the protein, but beyond that the small molecules or ions are likely to have little effect.

b. Chemical protectants

These include reductants such as BME and DTT, heavy-metal ion scavengers such as EDTA and EGTA, and compounds intended to prevent microbial infection such as sodium azide, phenol or chlorobutanol. Their effects are predictable, and their impact on the crystallization process is usually, but not always, of marginal significance.

c. Solubilizing agents and detergents

These include quaternary ammonium salts, sulfobetains, chaotropes such as urea and a range of surfactant and detergent molecules. It is accepted that detergents, principally non-ionic detergents such as β -octylglucoside (BOG) or dodecylmaltoside (DDM), need to be included in the mother liquors of membrane proteins to ensure their solubility and conformational integrity.

d. Poisons

They were originally employed to reduce twinning. These are generally low concentrations, 1–5%(w/v), of common organic solvents. They include compounds such as ethanol, DMSO, acetone, dioxane, butanol or MPD. They are likely to enhance the solubility of the proteins and slightly reduce the degree of supersaturation in the mother liquor, as well as lower the dielectric constant of the medium and more other effects.

e. Osmolytes, co-solvents and cosmotropes

They are compounds that exert their effects at relatively high concentrations, 1 M or more, and include a wide range of molecules such as sucrose, trehalose and other sugars, proline, TMAO, glycine, betaine, taurine, sarcosine, etc. The effect of their inclusion in the mother liquor is to stabilize (or destabilize) the native conformation of the protein by altering the interaction of the surface of the protein with water, or by altering the hydration layer and possibly the structured waters.

f. Multivalent molecules such as diamino or dicarboxylic acid containing molecules, or aliphatic moieties of various lengths carrying some combination of charged groups

This set of molecules might stabilize the conformations of proteins and reduce their dynamic character, by providing the proteins with small molecules that could reversibly cross-link charged groups (carboxyl and amino groups) on the surface of the protein, or form intramolecular hydrogen-bonding networks using surface polar groups. This potential mechanism of altering crystallization behavior, however, may indeed be pertinent.

g. Materials or compounds that somehow serve to enhance nucleation, including unique surfaces

These may include low concentrations of PEG, or other polymeric substances such as Jeffamine, emulsified in solutions of high salt concentration. The micro droplets of the polymeric phase serve to concentrate the protein locally and provide an

interface for nucleation to occur. This category should probably also include things like the gel used in cubic lipidic phase crystallization and surfaces which promote epitaxy and heterogeneous nucleation.

Unlike single component solutions for which it is possible to draw a phase diagram, the parameters affecting the crystallization of macromolecules solution are numerous, and we can only ever access an empirical phase diagram. On one hand, many aspects of the XRD technique are constantly improved: easier sample preparation and purification, better x-ray sources and detectors, robots for mounting crystals under the beam, streamlined data processing and validation, etc. On the other hand, the improvements around the crystallogenesis step focus on easing the trial-and-error approach: robots handling nanoscale volumes requiring less sample, diversification of the commercially available crystallization kits allowing a larger number of conditions to be tested, automated photography of the drops, image recognition of the drops' content, etc. The PDB is an incredibly rich source of data about successful crystallization parameters, as it contains information on crystallization conditions.

2.2 The Protein Data Bank(PDB)

The Protein Data Bank (PDB) is an open-access online repository containing information about solved proteins, DNA/RNA, and their complexes with metals and small molecules structures. Along with the atomic coordinates, associated metadata may include the sequence of these biological macromolecules and related experimental data and metadata., its species of origin and details of expression and structure determination [21, 22, 23]. From seven structures in 1971, the number of structures in the databank has grown to over 252 k nowadays. The predominant method of structure determination, accounting for over 80% of the entries, is XRD and the remaining 20% account for the Nuclear Magnetic Resonance spectroscopy (NMR), the 3D Electron Microscopy (3DEM), powder diffraction, fiber diffraction, etc.

2.2.1 The Metadata and the Experimental data in the PDB

The PDB powered by the RCSB PDB is committed to ensure that the PDB data is "FAIR" (Findable, Accessible, Interoperable and Reusable [24]. To ensure compliance with the FAIR Principles, the structures deposited are accompanied by comprehensive and standardized metadata which define a hierarchical description of each atom, specifying whether it belongs to a polymer (and, if so, to which residue), or corresponds to a metal ion, ligand, small-molecule solute, or water molecule. For macromolecules, additional annotations such as molecular name, source organism, and cross references to external bioinformatics resources are also included. Furthermore, information regarding the experimental method used for structure determination, as well as details on the preparation and characteristics of the sample, is systematically archived. Experimental data/metadata accompanying structures are processed using the universal OneDep system for deposition, validation and biocuration [25] www.wwpdb.org/documentation/biocuration. The wwPDB working group have adopted a common extensible PDBx/mmCIF data archiving framework with the associated mmCIF format as the master file format for the PDB Core Archive. All 3D structures in the PDB Core Archive are now stored and distributed in PDBx/mmCIF format [26].

In addition to PDBx/mmCIF, the wwPDB also distributes the atomic coordinates of every PDB structure in PDBML/XML format, which can be read using a standard XML parser (pdbml.pdb.org)[27] and as RDF (rdf.wwpdb.org/pdb) [28].

While the PDB infrastructure strongly supports compliance with the FAIR principles through standardized identifiers, metadata schemas, and interoperable file formats such as PDBx/mmCIF, the practical FAIRness of individual entries may vary depending on the completeness and consistency of depositor-provided annotations. In practice, important experimental or biological metadata are sometimes incomplete, inconsistently standardized, or embedded in free-text fields, which can limit machine readability and downstream reuse.

2.2.2 The PDB Application Programming Interfaces (APIs)

The RCSB.org website is powered by APIs which are exposed to be used in custom software or web-based applications by anyone around the world.

- Search API: serves to find out what identifiers match a certain search condition
- Data API: contains static data for each structure and serves to retrieve them when you know the PDB identifier.
- ModelServer API: is a service for accessing subsets of macromolecular model data
- VolumeServer API: is a service for accessing subsets of volumetric data
- Sequence Coordinates API: serves alignments between structural and sequence databases and integrates protein positional features from multiple resources
- Sequence Coordinates API: serves alignments between structural and sequence databases and integrates protein positional features from multiple resources

The two main APIs underlying the RCSB.org are the search API and data API as shown in figure 2.4.

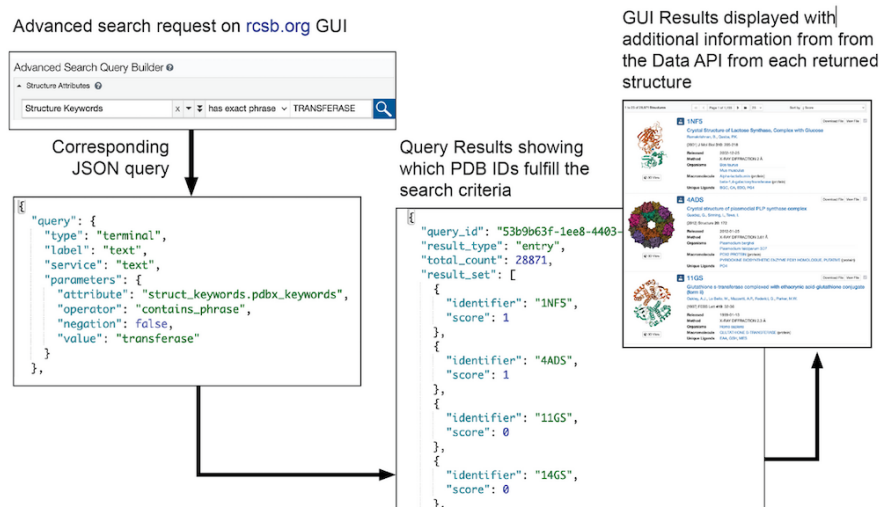


Figure 2.5: *RCSB PDB API Guide*

Search API

The Search API accepts HTTP GET or POST requests with JSON payloads. The base URI of search endpoint [RCSB Search API v2 Documentation](#). In GET method, search request should be sent as a URL-encoded query string in json parameter (<https://search.rcsb.org/rcsbsearch/v2/query?json={search-request}>). The search API is designed to return only the identifiers of relevant hits according to the Specified `return_type` of the identifiers (entry, `polymer_entity`, assembly, `non_polymer_entity`, `polymer_instance` and `mol_definition`) and additional metadata. The query context may consist of two types of clauses:

- Terminal node: performs an atomic search operation. example of the searching of a particular value in a particular field. Parameters in the terminal query clause provide match criteria for finding relevant search hits. The set of parameters differs for different search services.
- Group node: wraps other terminal or group nodes and is used to combine multiple queries in a logical fashion. The complex queries in the basic search can be built with "AND", "OR", operators as seen in figure 2.5.

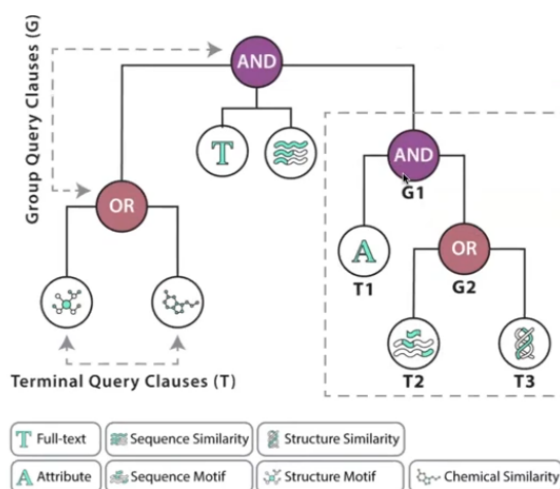


Figure 2.6: Terminal (T) and group (G) queries nodes in the search API

Data API

The RCSB PDB offers two ways to access data:

- REST-based API: It supports the HTTP GET method to access the PDB data through a set of endpoints (or URLs). See Data Organization section for more information on the underlying data organization. The path of the endpoints begins with <https://data.rcsb.org/rest/v1/core>, followed by the type of resource (e.g., `entry`, `polymer_entity`) and the corresponding identifier. Note that compound identifiers—such as entity ID, assembly ID, and entity instance ID (or chain ID)—are passed as path parameters (e.g., <https://data.rcsb.org/rest/v1/core/entry/4HHB>).
- GraphQL-based API: GraphQL server operates on a single URL/endpoint, <https://data.rcsb.org/graphql>.

[//data.rcsb.org/graphql](https://data.rcsb.org/graphql), and all GraphQL requests for this service should be directed at this endpoint.

The Protein Data Bank (PDB) provides a rich resource of experimentally validated crystallization conditions. The Advanced Search on RCSB.org is powered by the Search API, which translates user-defined criteria into a JSON query and returns a list of PDB IDs that satisfy those constraints. These results can then be explored through customizable layouts that retrieve additional annotations via the Data API. The bottleneck, therefore, lies in extracting this information and other useful metadata and transform them into a structured, human-readable format suitable for scientific analysis.

Chapter 3

Extraction of Macromolecules Crystallization Data Method

Mining crystallization cocktails can reveal valuable insights into the conditions that yield diffraction quality macromolecular crystals. However, despite the availability of such information across many deposited structures, analyzing it at large scale remains challenging due to its unstructured nature and the size of the dataset. To address this, modern data mining and big data methodologies have been developed, enabling efficient mapping, searching, and pattern analysis across large biological datasets. These approaches support various forms of analysis, including clustering, sequence homology, and structural comparisons, facilitating the extraction of meaningful knowledge from complex biological databases. In this chapter, we discuss the strategies implemented to develop the Crystallization Data Extraction (CDE) pipeline for macromolecule crystallization optimization.

3.1 General Pipeline from Macromolecule to Crystallization Cocktail

In this work, we designed a computational pipeline to automate the retrieval and standardization of crystallization metadata from the Protein Data Bank (PDB) repository. The workflow consists of four main steps, as illustrated in Figure 3.1. Each step is described in detail in the following sections.

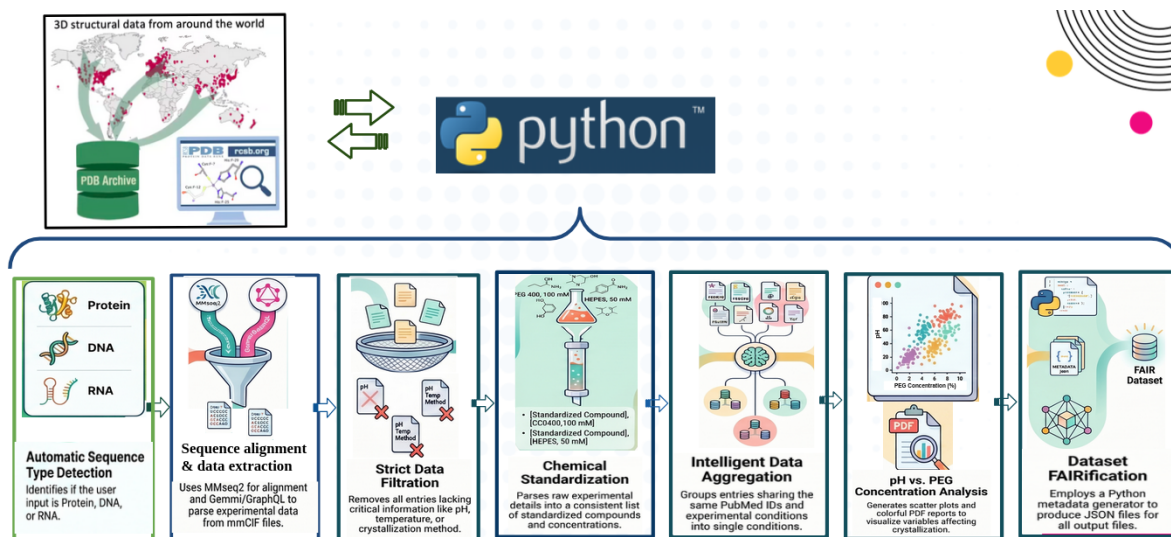


Figure 3.1: *Computational Pipeline automating the retrieval and standardization of experimental metadata from the PDB repository: from Macromolecule to Crystallization Cocktail*

3.2 The Sequence Homology Search

The PDB relies on several sequence alignment tools and pipelines to connect protein sequences with known 3D structures and to annotate entries consistently. These tools are used for simple alignment and also support structure validation, functional annotation, and cross referencing with other biological databases. The main sequence alignment approach used in the pdb is the pairwise sequence alignment methods that uses the Basic Local Alignment Search Tool (BLAST) [29, 30].

BLAST is used to compare a query sequence against sequences of known structures. It finds regions of local similarity between sequences and helps identify which known structures are similar to the queried sequence.

Similarly, the RCSB search API consolidates requests to heterogeneous search services among which the sequence. This service employs the MMseqs2 (Many versus Many Sequence Search) [31, 32] tool, which performs fast sequence matching searches based on the query sequence. MMseqs2 is actually one of the most important modern sequence alignment tools used in workflows related to the Protein Data Bank and is designed for ultra-fast, large-scale sequence searches and clustering. It achieves speeds orders of magnitude faster than tools like BLAST while maintaining comparable sensitivity.

The sequence homology search in this work consisted at extracting the sequence alignment details seen in figure 3.2 using a python 3 code `rscb_sequence_identity.py`.

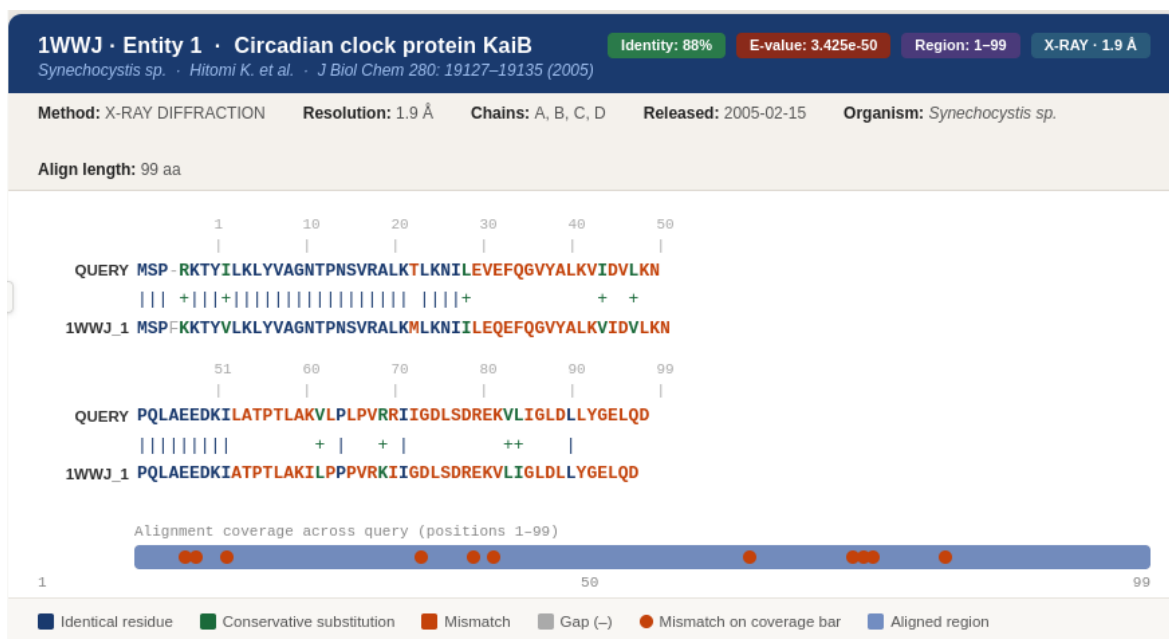


Figure 3.2: Sequence alignment results (sequence identity and E-value) from the advanced search of the RCSB PDB search API

In order to properly extract the necessary information (score, entity, sequence identity, and E-value) against the RCSB Protein Data Bank, we have typically used their sequence search API (MMseqs2-based). The detailed pipeline is described as follows:

- **Step1: Input** The sequence matching search that are available are the protein, DNA and RNA sequences.
- **Step2: MMseqs2 search**

A POST request is issued to [rcsb search website](#) using a sequence based scoring strategy, with all hits returned. The two main parameters are `E_value_cutoff`, which controls the statistical significance threshold, and `identity_cutoff`, which defines the minimum sequence identity. The API response consists of a ranked list of polymer entities, where the overall score appears at the top level, while detailed alignment information is nested within `services` → `nodes` → `match_context`.

- **Step 3: Filtration**

Each unique PDB ID is queried against the RCSB entry endpoint only entries for which `exptl[0][method]` contains "X-ray diffraction" are retained. To improve efficiency, results are cached per PDB ID so that each entry is requested only once. Entries determined by other experimental methods such as cryo-EM, NMR, electron diffraction, or neutron diffraction are automatically excluded.

- **Step 4: Field Extraction**

For every hit that passes the filter, the tool parses specific fields, extracting the PDB ID and entity from the identifier string, along with the score, sequence

identity, and E-value. The results are saved as `identity_minimal.csv` file in the output folder.

This search and filter phase feeds directly into the broader pipeline, where these filtered hits are merged with structural metadata to eventually produce the final crystallization condition reports.

Listing 3.1: *Python query for sequence based retrieval and X-ray filtering from the PDB*

```
RCSB_SEARCH_URL = "https://search.rcsb.org/rcsbsearch/v2/query"
RCSB_ENTRY_URL = "https://data.rcsb.org/rest/v1/core/entry"

def detect_seq_type(sequence: str):
    seq = sequence.upper()
    if set(seq) <= set("ACDEFGHIKLMNPQRSTVWY"):
        return "protein"
    elif set(seq) <= set("ATCG"):
        return "dna"
    elif set(seq) <= set("AUCG"):
        return "rna"
    else:
        return "protein"

def search_pdb_by_sequence(sequence, max_hits=1000):
    seq_type = detect_seq_type(sequence)
    target_map = { "protein": "pdb_protein_sequence", "dna": "pdb_dna_sequence", "
                  rna": "pdb_rna_sequence"}
    target = target_map[seq_type]
    query = {
        "query": {
            "type": "terminal",
            "service": "sequence",
            "parameters": {"target": target, "value": sequence}},
        "request_options": {
            "scoring_strategy": "sequence",
            "results_verbosity": "verbose",
            "paginate": {"start": 0, "rows": max_hits }},
        "return_type": "polymer_entity"}
    r = requests.post(RCSB_SEARCH_URL, json=query)
    r.raise_for_status()
    return r.json().get("result_set", [])

def is_xray(pdb_id: str) -> bool:

    try:
        r = requests.get(f"{RCSB_ENTRY_URL}/{pdb_id}", timeout=10)
        if r.status_code != 200:
            return False
        data = r.json()
        method = data.get("exptl", [{}])[0].get("method", "")
        return "X-RAY" in method.upper()
    except Exception:
        return False
```

3.3 Extraction of Structures Metadata from the PDB

Recently, researchers have designed the PDBx/mmCIF file format to store all data items and metadata relevant to macromolecular crystallographic experiments. The PDBx/mmCIF explicitly documents all relationships between common data items (e.g. atom and residue identifiers) which permits software applications to evaluate and validate referential integrity with any PDB entry, and maps information between the residue sequences of the experimental sample and the model coordinates. The effective

extraction of metadata into the crystallization cocktail involves three main stages. The metadata is deposited in the PDB by individuals and therefore is not uniform from one structure to another. For certain structures, all those information can be found in the mmCIF, and for some, those information can be found in the PDB json format or the graphQL tools. Figure 3.3 shows the general pipeline to extract those data and metadata from the mmCIF file.

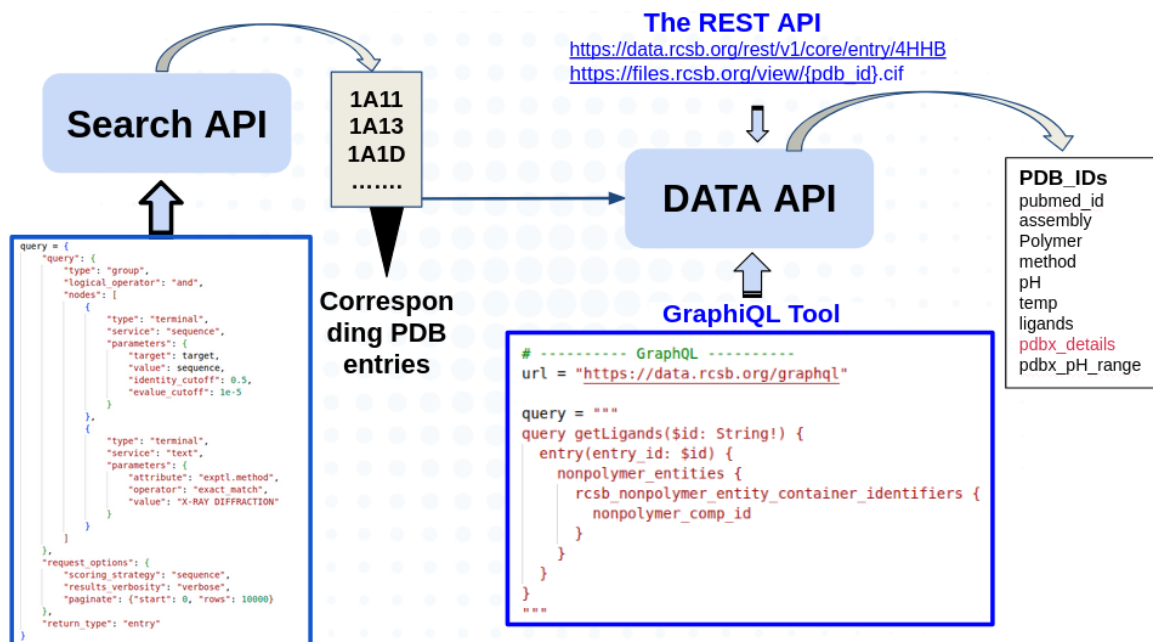


Figure 3.3: Overview of the RCSB Protein Data Bank (PDB) programmatic access workflow. The Search API is used to query the database using sequence and experimental criteria to retrieve corresponding PDB entries. Selected entries are then accessed through the Data API, REST API, or GraphQL interface to obtain the corresponding Metadata

The figure illustrates a workflow for retrieving and filtering structural data from the RCSB Protein Data Bank using multiple APIs. It begins with a query sent to the Search API, where sequence based parameters are defined to identify matching the structures and returns a list of corresponding PDB entries (e.g., 1A11, 1A13, 1A1D), which are then passed to the Data API for detailed information retrieval via REST endpoints.

In parallel, a GraphQL query is used to extract specific data such as ligand identifiers. The combined outputs provide structured metadata for each entry including PubMed ID, assembly, polymer type, experimental method, pH, temperature, and ligand details—forming a streamlined pipeline from search to data extraction.

Because of inconsistencies in the location of these metadata in the PDB repository, several functions were defined in the [PDB_searchAPI.py](#) python code to effectively extract the parameters. The code is organized into several key modules and functions that execute automated steps.

3.3.1 Input Processing and Sequence Search

- **detect_seq_type**

This function acts as a critical preprocessing component within the Sequence Input stage of the automated CDE pipeline. Its primary role is to determine whether a user-provided biological sequence corresponds to protein, DNA, or RNA by first normalizing the input to uppercase and then comparing its unique characters against predefined sets of valid biological symbols. Based on these comparisons, the function returns the labels "dna", "rna", or "protein"; notably, it is designed to default to "protein" if the sequence contains unrecognized or mixed characters. This automated categorization ensures that the correct sequence context is established before the pipeline proceeds to the RCSB MMseqs2 sequence search.

- **search_pdb_by_sequence**

The `search_pdb_by_sequence` function accepts a biological sequence and utilizes `detect_seq_type` to determine the appropriate search target (protein, DNA, or RNA) before the execution of a sophisticated automated query against the RCSB PDB Search API. This query is designed to fulfill two core project requirements: identifying homologous proteins through sequence similarity. To handle the high volume of data found in the global PDB Archive, the function employs a parallel extraction strategy using six concurrent workers to call `extract_mmcif_info`, which retrieves critical experimental parameters. Finally, the function consolidates these findings into a **tmp full.CSV** file, providing the structured historical data necessary to guide the design of new crystallization trials.

- **Parallel Execution**

It utilizes a "ThreadPoolExecutor" with six parallel workers to handle high volume data retrieval efficiently.

3.3.2 Optimization of Data Retrieval and Caching

To minimize redundant network traffic and ensure efficiency, the code implements local caching:

- **get_cached_mmcif**

The `get_cached_mmcif` function is a performance optimized retrieval utility that facilitates the collection of structural metadata and experimental parameters within the automated pipeline. By accepting a PDB ID as input, the function implements a local caching mechanism that first searches a designated directory for the corresponding mmCIF file the standard data format for the PDB; if the file is absent, it executes an automated GET request to the RCSB PDB server (files.rcsb.org) to download the necessary data.

This function is a critical component of the "Filter experimental conditions" and "Collect structural metadata" steps, as it provides the raw structural information

from which the tool extracts crystallization parameters. By saving a local copy of each retrieved file, the function ensures the pipeline is efficient and avoids redundant network traffic while identifying successful conditions for homologous macromolecules.

- **get_cached_pubmed_id**

The `get_cached_pubmed_id` function is a critical utility designed to fulfill the pipeline's requirement to group and count crystallization conditions at the publication level. By accepting a PDB ID as input, the function implements a local caching mechanism to store and retrieve citation metadata, ensuring that if a record is not already present, the tool executes an automated GET request to the RCSB PDB REST API to extract the unique PubMed identifier. This capability is essential for the data normalization phase, as it allows the tool to identify multiple structural entries originating from the same study, thereby preventing redundant data from artificially inflating trends in the final analysis of successful crystallization parameters

3.3.3 Crystallization Parameters Extraction and Text Mining

The core data-mining logic resides in a suite of functions that employ a two-tiered retrieval strategy:

- **Strategic extraction of specific data in the mmCIF file or `get_ph_from_mmcif_or_details`**

The functions `get_ph_from_mmcif_or_details`, `get_method_from_mmcif_or_details`, `get_temperature_from_mmcif_or_details`, `get_ph_range_from_mmcif_or_details` represent the core data mining logic within the “Filter experimental conditions” and “Collect structural metadata” stages of the automated pipeline.

Their primary objective is to transform pH, crystallization method, temperature, and pH range data from the PDB Archive into structured, machine-readable parameters. Each function employs a robust two-tier retrieval strategy: they first attempt to extract values from standardized mmCIF fields. If these fields are empty, they fall back to automated text mining using regular expressions applied to the free-text section `_exptl_crystal_grow.pdbx_details`. This approach directly implements the project's goal of interpreting descriptive annotations to ensure that no critical experimental information is lost.

By normalizing these variables for instance, ensuring temperatures are captured in Kelvin or formatting pH ranges as "low-high" strings, the functions enable the pipeline to consolidate historical success into the final CSV reports.

- **`fetch_ligand(pdb_id)`**

The function `fetch_ligands` is a specialized retrieval utility that implements an optional extension of the pipeline by identifying co-factors and bound molecules that may influence crystallization success.

To ensure thorough data collection from the PDB Archive, it employs a prioritized two-tiered strategy: first, it executes a POST request to the RCSB GraphQL API to retrieve non-polymer entity identifiers; if that fails, it falls back to automated parsing of the local or cached mmCIF file to extract component IDs from the `_pdbx_entity_instance_feature.comp_id` field.

This function directly supports the tool’s core objective providing researchers with a consolidated list of ligands associated with each successful trial.

- **`extract_mmCIF_info(pdbid)` function**

The function `extract_mmCIF_info` serves as the central data aggregation hub for the “Collect structural metadata” and “Filter experimental conditions” stages of the automated pipeline.

By accepting a PDB ID, it orchestrates the retrieval of multiple data points through coordinated calls to previously defined utility functions, such as `get_cached_pubmed_id` for publication tracking and `get_cached_mmCIF` for accessing raw structural data. The function leverages the `gemmi` library to parse mmCIF files and robustly extract assembly details and polymer types, thereby providing essential structural context.

There is a loop which aims to identify the oligomeric state (e.g., monomeric, dimeric) of the macromolecule by retrieving assembly identifiers and descriptive details from specific mmCIF fields, namely `_pdbx_struct_assembly.id` and `_pdbx_struct_assembly.oligomeric_details`. The logic is designed to prioritize the first descriptive detail found; however, if the PDB entry contains multiple biological assemblies, the code automatically appends a `"(*)"` symbol to the description to alert the researcher.

The `get_polymer_type_from_mmCIF` function enriches the metadata by determining the complexity of a protein structure, specifically distinguishing between single-polymer (`"uni_pol"`) and multi-polymer (`"complex"`) entries. By analyzing the `_entity.type` field within a retrieved mmCIF data block, the function counts the number of unique polymer entities to characterize the structural nature of the PDB entry.

This automated classification is an integral part of the “Enrichment & Analysis” phase, helping structural biologists better understand the broader context of successful crystallization conditions for homologous proteins.

This function also compiles a comprehensive dictionary of key experimental parameters including resolution, pH, temperature, and crystallization method while also incorporating ligands and cofactors as part of the pipeline’s optional extensions.

3.3.4 Data Refinement and Output

The `filter_experimental_conditions` function is a key computational step of the crystallization pipeline, specifically designed to refine the datasets retrieved during the

initial PDB search.

Its primary role is to transition the workflow from a **temporary full CSV** of raw structural hits to a **tmp filtered CSV** that contains only the most relevant experimental information.

The function reads an input CSV file and filters the data to retain only those rows where at least one of the critical experimental column: **pH**, **Temp**, **Method** or **pdbx_details** is not null. This filtering logic directly supports the tool’s core objective of consolidating experimental parameters to guide the design of new crystallization trials.

If no specific output name is provided, the function automatically generates a new filename with a **__filtered** suffix, saves the refined dataset to disk, and returns the absolute path of the new file for use in the next step of the pipeline.

This function acts as a quality control bridge within the workflow. It ensures that the subsequent **Add compounds** and **Merge RCSB + crystallization data** phases only process entries with actionable experimental data. By cleaning up the results at this stage, the tool prevents entries that lack crystallization metadata from biasing the final plots and merged reports (such as **{name}__merged_results.csv**) that are ultimately delivered to the user.

These functions collectively form an automated computational pipeline designed to transform a raw macromolecular sequence into a structured, machine-readable dataset of successful X-ray crystallization conditions. By integrating specialized tasks such as sequence categorization (**detect_seq_type**), parallelized PDB searching (**search_pdb_by_sequence**), and optimized data caching (**get_cached_mmcif**, **get_cached_pubmed_id**). The pipeline ensures efficient data retrieval.

The core logic relies on a robust two tiered extraction strategy (**extract_mmcif_info**), which precisely recovers experimental parameters like pH, temperature, and ligands from both standardized fields and descriptive annotations.

Finally, by refining the results through data validation (**filter_experimental_conditions**), these functions provide structural biologists with a reliable, data driven “shortcut” to guide the design of new crystallization trials.

3.4 Standardization of Crystallization details

3.4.1 Parsing Crystallization Details into a List of Compounds

Lynch et al. in 2020 [6] had developed a python based tool for a searchable and updatable database of crystallization conditions extracted from the free text field containing crystallization details found in the PDB entries. The database contains a method to parse the crystallization details in the PDB into a formatted list of compounds and their concentration, as well as a dictionary to standardize the names of the chemical compounds. This tool is freely available on [github](#).

The script `pdb_crystal_database.py` provides functionality for generating and querying the database of structures. The database is represented as a collection of `Structure` objects, which are serialized and stored in the file `structures.pkl`. This file constitutes the primary output of the script and serves as an essential input for the subsequent stages of the computational pipeline.

An updated version of the database was generated for this study by running the standalone `pdb_crystal_database.py` code in the `src` sub-folder. It aims to parse the inconsistently formatted crystallization details of the PDB structures based on XRD method into a consistently formatted list of compounds for computational analysis. This process was implemented using the `structure.parseDetails()` function, which extracts compound names, together with the `structure.standardizeNames()` function, which standardizes compound nomenclature based on a predefined dictionary containing possible representations of each compound.

The code was further modified to selectively extract information from structures determined using the XRD method. In addition, the pipeline was executed using the newer version of NLTK with the `tokenizers/punkt_tab` tokenizer, enabling the successful extraction of data from 195,183 structures.

Below is an example of the detail parsing process for one structure (PBD ID 3DZY):

```
15-18% PEG 3350, 25mM MgCl2, 100mM NH4Cl, 5mM DTT and 0.1M MES,
pH 6.5, vapor diffusion, hanging drop, temperature 277 K
```

After parsing details:

```
['PEG 3350', '16.5%', 'mgcl2', '25', 'nh4cl', '100', 'dtl', '5',
'mes', '100.0']
```

After standardizing names:

```
'PEG 3350', '16.5%', 'magnesium chloride', '25', 'ammonium
chloride', '100', 'DTT', '5', 'MES', '100.0'
```

3.4.2 Enrichment of Crystallization Details with the Standardized list of Compound

The python code `extract_structures.py` aims to enrich the experimental data with the 'add compound'. The code call three functions briefly described as follow:

- **`format_compounds(compound_list)`**

The python code `extract_structures.py` Converts a flat alternating list of compound names and concentrations into a readable string. Input `['PEG 4000', '10%', 'NaCl', '200']` becomes `"PEG 4000 (10%), NaCl (200)"`. Handles odd-length lists safely by appending the last compound without concentration, and returns an empty string for any invalid input.

- **`extract_peg_info(compound_str)`**

Scans the formatted compound string produced by `format_compounds` and extracts PEG-specific information. For each comma-separated component that contains the word "PEG", it pulls out the PEG number (e.g. 4000 from PEG 4000) and the numeric concentration value stripped of units or percentage signs. Multiple PEG entries are joined with semicolons. Returns two strings: `PEG_Id` and `PEG_con`. Returns empty strings if the input is not a valid string.

- **append_compound_to_filtered_csv** The main function that loads a pickle file containing a list of `Structure` objects using a fake module injection (`types.ModuleType`) to allow unpickling without the original class definition. It auto-detects which attribute holds the PDB ID and which holds the compound list by scanning the attribute names of the first object. It then builds a lookup dictionary mapping each PDB ID to its formatted compound string, maps this to the filtered CSV dataframe by adding a new column titled "Compounds(con_unit=mM)", calls `extract_peg_info` to add the `PEG_Id` and `PEG_con` columns, and saves the result to `output_csv_file`.

3.5 Visualization and Reporting

The `run_plot` function of the python code `plot.py` constitutes the final phase of the automated crystallization workflow. It receives the merged CSV file path and produces three output files: two high-resolution scatter plots and one colored PDF table. The function operates in two sequential stages (figure 3.4).

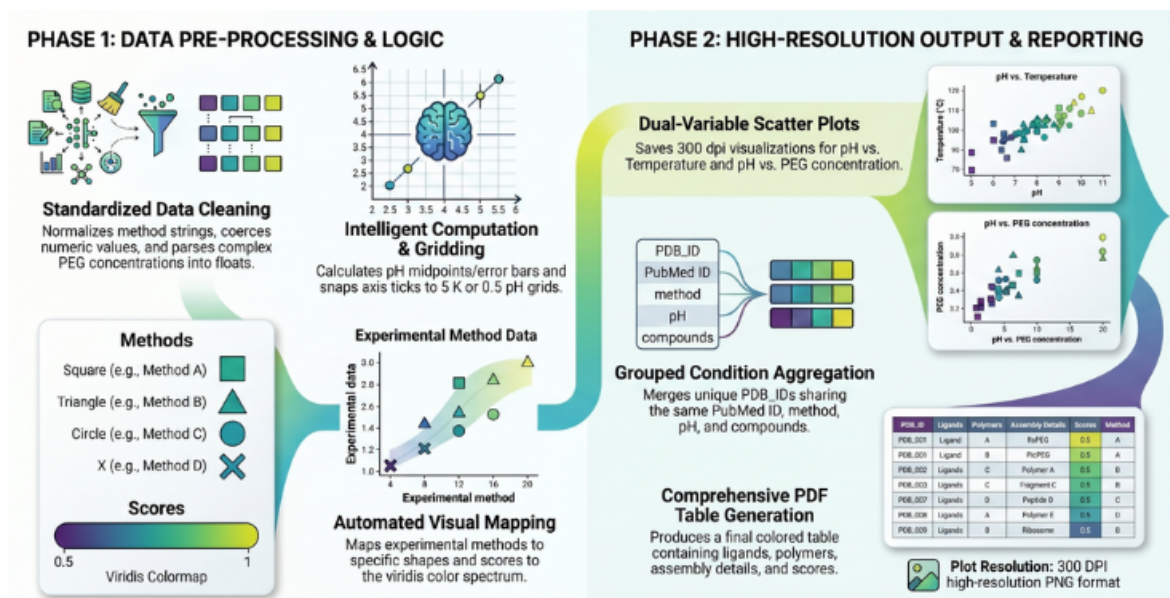


Figure 3.4: *visualization and reporting Pipeline*

3.5.1 Data Preprocessing

- **Data preparation and cleaning** The merged CSV is loaded with `pd.read_csv()`. The pH, Temp, and Score columns are coerced to numeric using `pd.to_numeric(...)`.

`errors="coerce").` Score is clipped to `[0.5, 1]` using `df["Score"].clip(0.5, 1)`.

- **Visual parameter assignment** `assign_marker(Method)` maps crystallisation methods to markers: square (s) for hanging drop, triangle ($\hat{}$) for sitting drop, circle (o) for vapor diffusion, and X for unspecified methods. Fallback markers (D, v, P, *) are assigned for unrecognised methods. Score is mapped to color using the `viridis` colormap.
- **pH computation and error bars** `parse_ph_range(ph_range)` extracts bounds from `pdbox_ph_range` using regex. `compute_plot_ph(row)` converts each row into a plotting pH value. Single values use zero error and the ranges are converted to midpoints with $\pm$ half-range error bars. Missing values are set to `np.nan` and flagged via `df["has_ph"]`.
- **Coordinate handling for Missing Data** The missing temperature values are assigned `no_temp_x = temp_min - 4`. The missing pH values are assigned `no_ph_y = ph_min - 0.5`. Off-scale points are placed in "No Temp" and "No pH" regions. Separation is visualized using `ax.axhline()` and `ax.axvline()`. The PEG concentration is parsed via `parse_peg_con(x)` and stored in `df["PEG_con_plot"]`.

3.5.2 High Resolution Output and Reporting

Once the data are processed, the pipeline generates high-quality visual and tabular summaries:

- **pH vs Temperature:** Display temperature in Kelvin with pH error bars, individual PDB_ID labels, and a score colorbar.
- **pH vs PEG concentration:** Illustrates pH versus PEG concentration percentage, utilizing a "No PEG" sentinel for entries lacking that specific precipitant.
- **Condition Grouping Logic:** To condense the final report, the system merges PDB entries that share identical publication IDs, methods, pH values, and chemical compounds into single cells.
- **Colored PDF table:** A matplotlib table is created using `ax.table()` on an 18×10 inch figure. It includes grouped headers and sub-headers. Columns include PDB_ID, Score, Seq_id, Pubmed_id, Polymer, Assembly, Method, Ligands, pH, Temp, and Compounds(con_unit=mM). Text wrapping is applied using `textwrap.wrap()` and the final PDF is saved as `{protein_name}_Cryst_cocktail_Table.pdf`.

The python codes are implemented through the `main.py` following the workflow described below in figure 3.5.

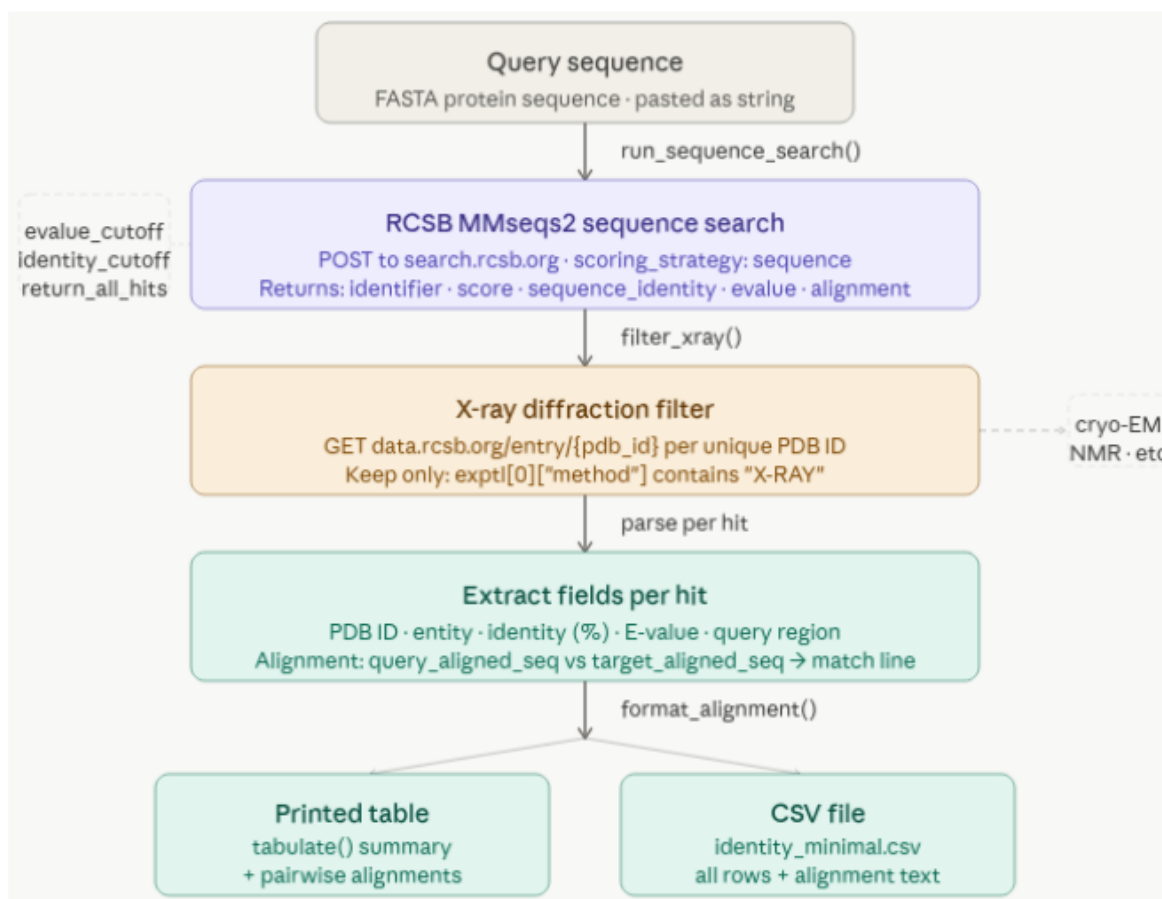


Figure 3.5: *The Main.py code execution pipeline*

the execution of the pipeline through the code `Main.py` returns the sequence saved as fasta file and gives the results in csv, pdf and png files in an output folder. These outputs will be presented and discussed in the next chapter. Additionally, scripts taking as input a fasta file with a set of homologue sequences are provided and will be further presented in appendix B. A web page implementing this pipeline was developed using Django and will be presented in chapter 4.

Chapter 4

Results and Discussions

This chapter presents the main output of the code developed throughout the project with the purpose of optimizing macromolecular crystallization process.

4.1 PDB Extraction with Sequence Alignment Score

The extracted metadata related to the sequence alignment are presented in figure 4.1 below.

The PDB ID (Protein Data Identifier) is a unique four character code assigned to each 3D structure deposited in the PDB. It is essentially the "accession number" for a molecular structure, typically a protein, nucleic acid, or complex. It is generally used to search for detailed structural data sequence search API (MMSeq2-based alignment).

The sequence identity is a log-odds alignment score that reflects how good the match is between the query sequence and a PDB sequence. This is influenced by the number of identical residues, conservative substitutions, gaps, alignment length, and statistical weighting of the substitution matrices. The higher score means better alignment (strong structural similarity).

The similarity score is used most effectively to rank hits relative to one another within the same search session, rather than for absolute biological interpretation; instead, the E-value and sequence identity should be prioritized for final homology assessment. In the context of the RCSB advanced sequence similarity search, sequence identity is defined as the percentage of characters (residues) that match exactly between the query sequence and the target sequence over the length of the alignment. These metrics are essential for the pipeline to identify true homologous proteins, ensuring that the historical experimental parameters retrieved from the PDB are biologically relevant to the user's protein of interest. Figure 4.1 presents an example of the alignment of a kaiB protein '4KSO' with '15RP'. The red dots visible along the alignment bar indicate the position where the two sequences differ-mismatches or substitutions. The purple bar below represents the aligned regions.

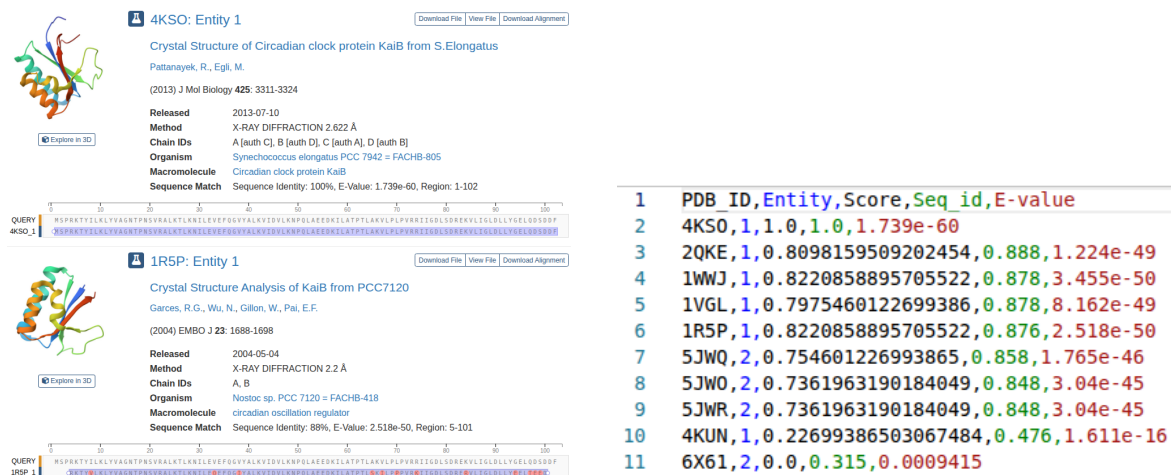


Figure 4.1: Example of the sequence similarity search results of the *kaiB* (4KSO vs 4KSO and 4KSO VS 15RP) from the RCSB website (left) and the specific data extraction using MMseqs2 (right)

The figure shows the alignment with the sequence identity 89% accross the entire 99-residue region with an E-value of 1.21 e-49, making this entry a strong candidate both as a molecular replacement model and as a crystallization condition reference. The *Entity* column denotes the entity identifier (e.g., 4KSO_1) within a PDB entry corresponding to the polymer that matches the query sequence.

This identifies the specific macromolecular chain whose amino acid sequence is homologous to the input sequence. Given this level of sequence identity, the conditions used to determine the corresponding PDB structure will likely be a good starting point to crystallize the query macromolecule

4.2 Extrated Metadata for Crystallization Optimization

After identifying PDB entries through sequence alignment, the results are consolidated into a final merged CSV file that includes both structural metadata and crystallization experiment data. Figure 4.2 presents an excerpt from the *kaiB_merged_results.csv* file generated by the execution of the pipeline. It includes all PDB hits corresponding to the *kaiB* sequence (e.g., 4KSO) with a score greater than or equal to 0.5.

1	PDB_ID	Entity	Score	Seq_id	E-value	Resolution	Pubmed_id	Method	pH	Temp	Ligands	Polymer	Assembly	pdbx_ph_range	pdbx_details	Compounds (con unit=mM)	PEG	ID	PI
2	926B	1	0.0	1.0	8.179e-221	2.7	41557701	0	"VAPOR DIFFUSION, SITTING DROP"	8.2	277.15	"ZN, AIC09"	uni_pol,monomeric	"1.8 M sodium/potassium phosphate pH 6.5"					
3	6WUJ	1	0.0	9935691318327974	0.996	2.621e-219	2.79	33067239	0	"VAPOR DIFFUSION, SITTING DROP"	291.0	"ZN, MG"	complex,dimeric(*)	"Ammonium Sulfate"	ammoniu				
4	926C	1	1.0	0.0	8.179e-221	2.06	41557701	0	"VAPOR DIFFUSION, SITTING DROP"	290.15	"ZN, CL, AIC1A"	uni_pol,monomeric(*)	"3.5 M Sodium formate"	sodium for					
5	6MX4	1	0.0	9935691318327974	0.996	2.621e-219	1.655	33067239	0	"VAPOR DIFFUSION, SITTING DROP"	8.5	291.0	"ZN,complex,dimeric"	"0.8 M Potassium sodium tartrate"					
6	7CMB	1	1.0	0.0	8.179e-221	2.59	32895623	0	"VAPOR DIFFUSION, HANGING DROP"	5.6	296.0	"TTT, ZN"	uni_pol,monomeric(*)	"0.24M ammonium acetate"	0.1M sodium				
7	9BRX	1	0.0	9935691318327974	1.0	3.592e-219	1.8	39140855	0	"VAPOR DIFFUSION, HANGING DROP"	291.0	"S04, ZN, AIASL"	uni_pol,monomeric	"0.2M sodium citrate"					
8	9H1X	1	0.0	9935691318327974	1.0	3.592e-219	2.7		"VAPOR DIFFUSION, HANGING DROP"	8.0	293.15	"GOL, ZN, CL, S04"	uni_pol,dimeric	"1.45 M Ammonium Sulfate"					
9	9FTF	1	0.0	9919614147989968	0.996	6.747e-219	1.7		"VAPOR DIFFUSION, SITTING DROP"	6.0	288.0	"ZN, MLI, PEG, NA, CL"	uni_pol,monomeric	"1.8 M sodium malonate"					
10	7CJD	1	0.0	9919614147989968	0.996	6.747e-219	2.501	32895623	0	"VAPOR DIFFUSION, HANGING DROP"	8.1	296.0	"ZN, EDO"	uni_pol,tetrameric	"3dextran sulfate"				
11	8EUA	1	0.0	9935691318327974	0.996	2.621e-219	3.1	36977673	0	"VAPOR DIFFUSION, SITTING DROP"	289.0	"WUK, ZN, S04"	uni_pol,monomeric	"PEG 3350, CaCl2, Cd"					
12	9F7G	1	0.0	9919614147989968	0.996	6.747e-219	2.3		"VAPOR DIFFUSION, SITTING DROP"	6.0	288.0	"ZN, GOL, S04"	uni_pol,monomeric	"2.5 M ammonium sulfate"	0.1M				
13	9H1G	1	0.0	9935691318327974	1.0	3.592e-219	1.95		"VAPOR DIFFUSION, HANGING DROP"	8.0	293.15	"CL, ZN, GOL"	uni_pol,dimeric	"1.45 M Ammonium Sulfate"	0.1M				
14	9F7H	1	0.0	9919614147989968	0.996	6.747e-219	1.5		"VAPOR DIFFUSION, SITTING DROP"	4.5	288.0	"ZN, GOL, P04"	uni_pol,monomeric	"1.2M K2HPO4, 0.8M NaH2PO4"					
15	6WZU	1	0.0	9903536977491961	1.0	9.246e-219	1.79	33531496	0	"VAPOR DIFFUSION, HANGING DROP"	4.5	277.0	"ZN, GOL, P04, CL"	uni_pol,monomeric	"0.1 M acetate"				
16	8UUG	1	0.0	9903536977491961	1.0	9.246e-219	2.74	38547259	0	"VAPOR DIFFUSION, SITTING DROP"	6.3	277.0	"XXW, GOL, ACT, ZN, CL"	uni_pol,monomeric	"5.5-6.5"	"0.1 M MES buffer"			
17	7JN2	1	0.0	9903536977491961	1.0	9.246e-219	1.93		"VAPOR DIFFUSION, HANGING DROP"	6.0	277.0	"Y41, ZN, CL, ACT, UNX"	uni_pol,monomeric	"0.1 M MES buffer"					
18	9DNV	1	0.0	9903536977491961	1.0	9.246e-219	2.4	39948104	0	"VAPOR DIFFUSION, SITTING DROP"	5.7	277.0	"CL, ZN, ACY, A1BEH"	uni_pol,monomeric	"5.5-6.5"	"200 mM"			
19	7J1W	1	0.0	9903536977491961	1.0	9.246e-219	2.3	33531496	0	"VAPOR DIFFUSION, HANGING DROP"	6.0	277.0	"VBY, ZN, CL, MES"	uni_pol,monomeric	"0.1 M MES buffer"				
20	7RZC	1	0.0	9903536977491961	1.0	9.246e-219	2.04		"VAPOR DIFFUSION, SITTING DROP"	7.5	277.0	"JWX, ZN, CL, EDO"	uni_pol,monomeric(*)	"0.2 M magnesium acetate"					
21	8UUV	1	0.0	9903536977491961	1.0	9.246e-219	3.01	38547259	0	"VAPOR DIFFUSION, SITTING DROP"	6.4	277.0	"Y21, ZN, CL"	uni_pol,monomeric	"5.5-6.5"	"0.2M Zinc Acetate"			
22	7SDR	1	0.0	9903536977491961	1.0	9.246e-219	2.72		"VAPOR DIFFUSION, SITTING DROP"	7.5	277.0	"JW9, ZN, EDO, CL, GOL"	uni_pol,monomeric(*)	"0.2 M sodium citrate"					
23	7SGW	1	0.0	9903536977491961	1.0	9.246e-219	1.95		"VAPOR DIFFUSION, SITTING DROP"	6.0	277.0	"CL, ZN, L30"	uni_pol,monomeric	"0.1 M Bis tris propane, pH 7.0"					
24	8UUV	1	0.0	9903536977491961	1.0	9.246e-219	3.01	38547259	0	"VAPOR DIFFUSION, SITTING DROP"	5.9	277.0	"XYR, ACT, ZN, CL, GOL"	uni_pol,monomeric	"5.5-6.5"	"0.2 M MES buffer"			
25	8UUF	1	0.0	9903536977491961	1.0	9.246e-219	2.84	38547259	0	"VAPOR DIFFUSION, SITTING DROP"	6.4	277.0	"XW0, ACT, ZN, CL"	uni_pol,monomeric	"5.5-6.5"	"0.2M Zinc Acetate"			
26	9DNV	1	0.0	9903536977491961	1.0	9.246e-219	2.3	39948104	0	"VAPOR DIFFUSION, SITTING DROP"	5.6	277.0	"A1BEG, ACT, ZN, CL, EDO"	uni_pol,monomeric	"5.5-6.5"	"0.2 M MES buffer"			
27	9D0I	1	0.0	9903536977491961	1.0	9.246e-219	2.3	39948104	0	"VAPOR DIFFUSION, SITTING DROP"	5.8	277.0	"ZN, CL, ACT, PEG, A1BEL"	uni_pol,monomeric	"5.5-6.5"	"0.2 M MES buffer"			
28	7KOL	1	0.0	9903536977491961	1.0	9.246e-219	2.58		"VAPOR DIFFUSION, HANGING DROP"	6.0	277.0	"Y96, ZN, CL, MES"	uni_pol,monomeric	"0.1 M MES buffer"	0.2 M				
29	8UUV	1	0.0	9903536977491961	1.0	9.246e-219	2.8	38547259	0	"VAPOR DIFFUSION, SITTING DROP"	6.0	277.0	"CL, ZN, PEG, XYI"	uni_pol,monomeric	"5.5-6.5"	"0.2M Zinc Acetate"			
30	8UOB	1	0.0	9903536977491961	1.0	9.246e-219	2.52	38547259	0	"VAPOR DIFFUSION, SITTING DROP"	6.2	277.0	"XB5, ZN, CL"	uni_pol,monomeric	"5.5-6.5"	"Zinc Acetate"			

Figure 4.2: This is an excerpt of a protein PLpro csv providing key structural metadata associated with each hit, including **pdb_id**, entity number, score, sequence identity, resolution, **Pubmed_id**, experimental method, ligands, assembly, polymer, pH, temperature, **pdbx_ph_range**, and **pdbx_details**.

Each of these parameters, extracted from the PDB, plays a distinct and complementary role in the crystallization optimization process. Together, these data transform a PDB hit from a simple sequence similarity result into an actionable experimental framework to guide crystallization trials.

4.2.1 Pubmed_id

The Pubmed ID provides a critical link between raw crystallization data and the underlying scientific rationale. Although a PDB entry records the experimental conditions used, the associated publication explains the reasoning behind these choices, including protein behavior during trials, unsuccessful attempts, and key factors leading to success.

This information is particularly valuable in the crystallization optimization process, as the methods section often details the full screening strategy such as initial sparse matrix screens, effective additives, protein concentration adjustments, and the use of techniques such as seeding.

4.2.2 Polymer type

Whether the entity is a protein, RNA, DNA, or a protein nucleic acid hybrid, the polymer type defines the entire chemical framework within which crystallization occurs. This matters for optimization because the precipitant chemistry that works for pure proteins differs substantially from those that work for RNA or DNA - protein complexes.

For example, if the hit is a protein-RNA complex and your target is a pure protein, the weight of the non-nucleic-acid conditions is more heavily, and any polyamine specific additives should be ignored. Therefore, Distinguishing between single-polymer ("uni_pol") and complex structures helps researchers understand if a successful condition was tailored for an individual protein or a multi-component assembly, which often requires different stabilization strategies.

4.2.3 Assembly

The biological assembly field describes how many copies of the entity and which combination of entities constitute the functional unit: monomer, homodimer, heterotetramer, etc. This has direct consequences for crystallization because the oligomeric state of the protein in solution determines which surfaces are available to form crystal contacts.

A protein that is a dimer in solution cannot form the same lattice contacts as a monomer because the dimer buries one entire face of each subunit at the dimer interface, leaving only the remaining surfaces free to participate in crystal packing.

The tracking of the oligomeric state provides a complete picture of the physical state of the proteins in solution. Because protein behavior and crystal contact formation are highly dependent on oligomerization, this context is crucial for choosing the right starting parameters.

4.2.4 Ligands

Ligands, including cofactors, metal ions, inhibitors, substrate analogs, and nucleotides are key determinants in protein crystallization, as they directly influence the conformational state of the protein. They can stabilize specific conformations and promote crystal packing through intermolecular contacts.

The Knowledge of ligands present in successful crystallization trials provides a rational basis for optimization, as including equivalent molecules (e.g: inhibitors, metal ions, or nucleotide analogs) can enhance crystal formation. In particular, metal ions such as zinc, calcium, and magnesium often facilitate crystal contacts through coordination with surface residues.

4.2.5 Method

vapor diffusion hanging drop, sitting drop, batch, micro-batch under oil and dialysis free interface diffusion define the physical mechanism by which supersaturation is achieved and the kinetics of how the system traverses the phase diagram. Vapor diffusion is the dominant method because it allows for slow and controlled equilibration, but the choice between hanging and sitting drop affects drop stability, evaporation rate, and ease of seeding.

Knowing the specific technique is necessary for reproducibility and the selection of the appropriate instrumentation, such as automated dispensers or monitoring systems.

4.2.6 Crystallization Cocktail

These parameters form the core "recipe" for successful trials and are the primary variables analyzed.

- **pH:** determines the net surface charge of the protein, which in turn controls the balance between electrostatic repulsion that keeps the protein in solution and the weaker attractive interactions that drive nucleation and crystal contact formation.

The pH extracted from a homologous structure gives the scientist a starting point that is already known to be compatible with a similar protein surface.

- **Temperature:** affects both protein solubility and the kinetics of nucleation and growth. Many proteins that aggregate at room temperature crystallize cleanly at 277K because a lower temperature reduces the rate of conformational fluctuation and slows aggregation relative to ordered nucleation. The temperature recorded in the PDB entry tells which thermal regime the experimenter found productive and whether a cold room or incubator setup is required.
- **pdbx_details:** is the free-text equivalent in the mmCIF file and is often the most information dense field. It typically contains details not captured in structured fields, such as the presence of additives (e.g: detergents), reducing agents (e.g., DTT or TCEP), cryoprotectants, seeding protocols, protein concentration, reservoir-to-drop ratios, and qualitative descriptions of crystal morphology.

It may also report experimental nuances, such as extended growth times or the critical role of a specific additive. Careful examination of the `pdbx_details` field for high identity hits beyond simply extracting numerical parameters such as pH and temperature often reveals key insights that can determine the success or failure of crystallization trials.

A new column was created in the csv file *Compounds* (`con_unit = mM`) to report the list of standardized chemical compound and their concentration was extracted from `pdbx_details` for further analysis.

The availability of this information allows the scientists to have the reasoning of another crystallographer who has already solved the same problem for a closely related protein.

4.3 Data Analysis

In order to have condensed readable results that can be directly exploited by scientists, we have merged PDB entries that share identical publication IDs, methods, pH values, and chemical compounds into single cells. These data are inserted into one line and we have produced a colorful table that will be directly exploitable by scientists. Figure 4.3 and 4.4 show the pdf table generated when executing the code with kaiB (4KSO) and PLpro (13MI) proteins sequence respectively. This technique was also applied for the plot of the PDB ID distribution of method, pH vs PEG concentration (%). Empty entries in the dataset indicate missing information in the PDB, either because those details were not reported during deposition or were not part of the crystallization conditions used.

										CRYSTALLIZATION COCKTAILS
PDB_ID	Score	Seq_id	Pubmed_id	Polymer	Assembly	Method	Ligands	pH	Temp	Compounds(con_unit=mM)
4KSO	1.000	1.000	23796516	uni_pol	tetrameric	vapor diffusion		5.7	292.0	sodium acetate (100.0), sodium formate (500.0), glycerol (5% v/v)
1RSP	0.822	0.876	15071498	uni_pol	dimeric(*)	vapor diffusion hanging drop		7.0	298.0	PEG 400 (None)
1WWJ	0.822	0.878	15716274	uni_pol	tetrameric	vapor diffusion	BET, IMD, MLT	6.0	298.0	betaine (None)
2QKE	0.810	0.888	18497745	uni_pol	tetrameric(*)	vapor diffusion hanging drop		5.0	293.0	PEG 3350 (10%), DMSO (10%), sodium acetate (100.0)
1VGL	0.798	0.878	16227211	uni_pol	tetrameric(*)	vapor diffusion hanging drop	HG	4.5	293.0	PEG 3350 (None), DMSO (None), sodium acetate (None)
5JWQ	0.755	0.858	28302851	complex	dodecameric	vapor diffusion hanging drop	ADP	5.75	295.0	sodium citrate (100.0), PEG 3000 (22%)
5JWR	0.736	0.848	28302851	complex	tetrameric(*)	vapor diffusion hanging drop	PO4	7.0	295.0	ammonium sulfate (150.0), tris (100.0), PEG 4000 (10%)
5JWO	0.736	0.848	28302851	complex	dimeric	vapor diffusion hanging drop	ADP		295.0	potassium acetate (200.0), PEG 3350 (10% w/v), glycerol (2.5%)

Figure 4.3: Colored pdf table presenting necessary metadata extracted for part the PDB_ID identified having similar sequence with the kaiB 4KSO protein

										CRYSTALLIZATION COCKTAILS
PDB_ID	Score	Seq_id	Pubmed_id	Polymer	Assembly	Method	Ligands	pH	Temp	Compounds(con_unit=mM)
1V0H, 1OAF, 1OAG, 2CL4, 2XJ6, 2XI6, 2XIF, 2XIH	1.000	1.000		uni_pol	monomeric	unspecified	HEM, NA, SHA	8.3		lithium sulfate (2250.0), HEPES (100.0)
2GHH, 2GHH, 2GHE, 2GHD, 2GHC	1.000	1.000		uni_pol	monomeric	vapor diffusion sitting drop	K, NO, HEM	8.3	273.0	HEPES (100.0), lithium sulfate (2250.0)
2VCF, 2VCS	1.000	1.000		uni_pol	monomeric	unspecified	ISZ, HEM, NA	8.3		lithium sulfate (2500.0)
7BI1, 5JQR	1.000	1.000		uni_pol	monomeric	vapor diffusion hanging drop	HEM, K	8.3	293.0	lithium sulfate (2250.0), HEPES (100.0)
5JPR	1.000	1.000		uni_pol	monomeric	vapor diffusion hanging drop	SO4, K, HEM	8.0	300.0	lithium sulfate (2250.0), HEPES (100.0)
6TAE, 6XV4	1.000	1.000		uni_pol	monomeric	vapor diffusion hanging drop	HEM, SO4		300.0	lithium sulfate hepes (None)
2VO2, 2VCN, 2VNZ, 2VNX	0.988	0.996		uni_pol	monomeric	unspecified	HEM, NA, SO4	8.3		
2GGN	0.988	0.996		uni_pol	monomeric	vapor diffusion sitting drop	SO4, HEM, NA	8.3	273.0	lithium sulfate (2400.0), HEPES (100.0)
2WD4	0.988	0.996		uni_pol	monomeric	vapor diffusion sitting drop	FE, SO4, TBV, NA	8.3		lithium sulfate (2250.0), HEPES (100.0)
3ZCG	0.980	0.992		uni_pol	dimeric	vapor diffusion sitting drop	HEM, SO4, K, EPE	8.5		lithium sulfate (2500.0), HEPES (100.0)
3ZCH	0.980	0.992		uni_pol	dimeric	unspecified	HEM, SO4, EPE, K	8.5		lithium sulfate (2500.0), HEPES (100.0)
9THS	0.947	1.000		uni_pol	monomeric	vapor diffusion hanging drop	HEM, CA	6.0	298.7	tris-HCl (100.0), magnesium sulfate (2700.0)
7S10	0.935	0.987		uni_pol	monomeric	vapor diffusion hanging drop	HEM, K, SO4	8.0	293.0	magnesium sulfate (None)
9KOE	0.919	0.983		uni_pol	monomeric(*)	vapor diffusion sitting drop	PO4, HEM	8.5	289.0	
5LB6	0.919	0.983		uni_pol	monomeric(*)	vapor diffusion sitting drop	SO4, HEM	5.5	293.0	bis-tris (100.0), ammonium sulfate (2000.0)
8FF6, 8FF7	0.813	0.862		uni_pol	monomeric(*)	vapor diffusion hanging drop	HEM, NA	8.5	277.0	lithium sulfate (200.0), tris (100.0), PEG 4000 (30%)
8DJX, 8DJT, 8DJS	0.809	0.854		uni_pol	monomeric	vapor diffusion sitting drop	SO4, HEM, NA	8.5	277.0	lithium sulfate (200.0), tris (100.0), PEG 4000 (30%)

Figure 4.4: Colored pdf table presenting necessary metadata extracted for part the PDB_ID identified having similar sequence with the kinase protein 1OAF. It also shows how the hits having the same Pubmed_id and crystallization cocktail information are combined in one line (see as one condition)

Figure 4.5 summarizes the scatter plots of the experimental method of four (04) macromolecules which are: the proteins PLpro (13MI), KaiB (4KSO), kinases (1OAF) and RNA (5DNO). The plot is the pH against PEG concentration (%) and Each point corresponds to a successful crystallization condition, with marker shapes denoting the

experimental method (e.g., vapor diffusion, microbatch) and color indicating a crystal quality score.

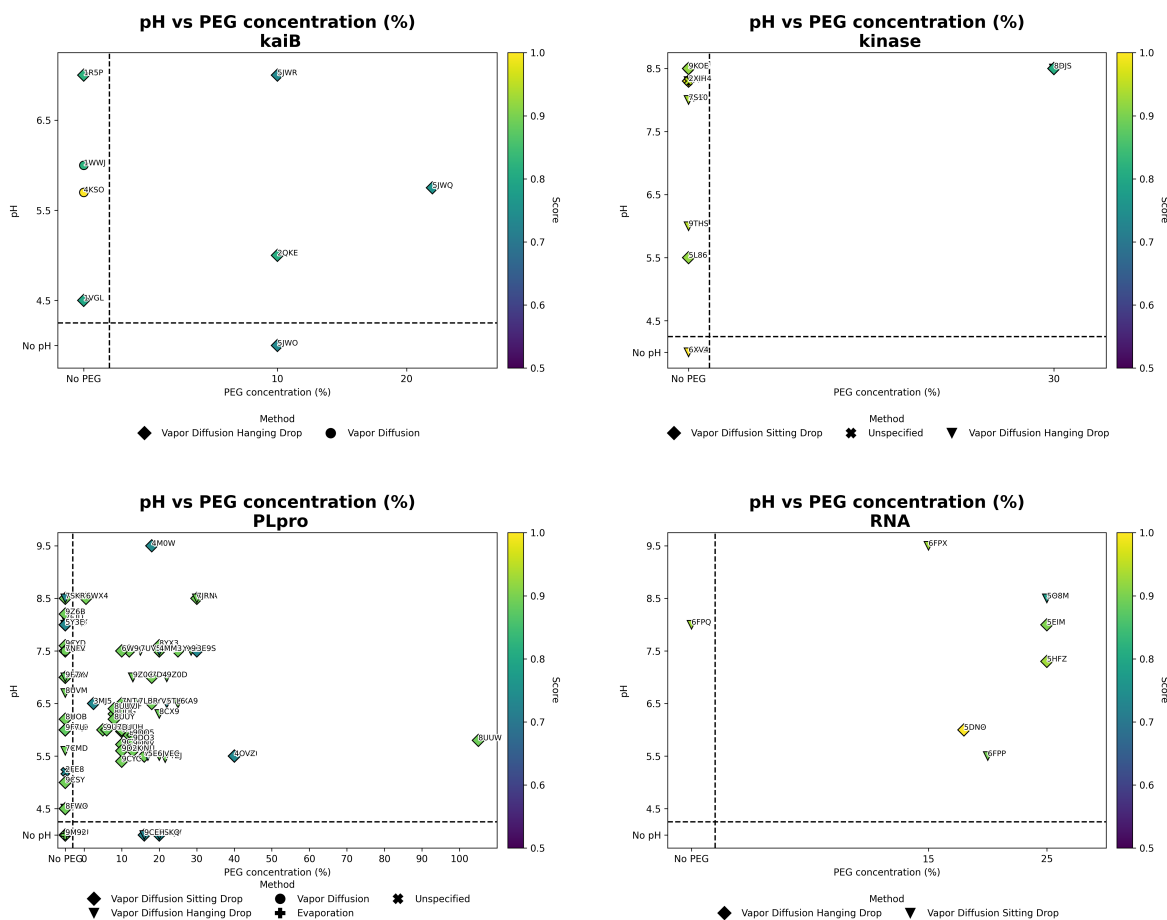


Figure 4.5: *Crystallization method scatter plots of pH versus PEG concentration across four macromolecular classes: KaiB, kinase, PLpro, and RNA.*

Across all panels, successful crystallization spans a broad pH range (4 – 10), each system exhibits clear clustering within specific regions. PEG concentrations also vary widely, with many successful conditions between 10–30%, alongside a few outliers at much higher PEG levels.

The dashed lines denote boundary conditions (e.g: absence of PEG or undefined pH), emphasizing that most successful crystallization outcomes depend on the interplay of both parameters. Distinct clustering patterns are observed across protein classes, and the number of recorded hits varies substantially: 8 for KaiB, 292 for PLpro, 304 for kinases, and 79 for RNA.

The figure illustrates that there is no universal condition each protein type samples a different region of pH–PEG space. However, it also reveals common “hotspots” (e.g., moderate PEG, near-neutral pH) that are good starting points for the screening. The spread of points emphasizes the need for broad initial screens, followed by focused optimization around promising clusters.

The variation in methods suggests that screening multiple crystallization techniques

can improve success rates. Overall, it supports a data-informed approach: uses known successful condition distributions to guide initial screens, then refine locally to improve crystal quality and reproducibility.

4.4 Crystallization Cocktail Webpage

To facilitate access to metadata extracted by the pipeline (figure 4.6), a web application was developed using the Django framework.

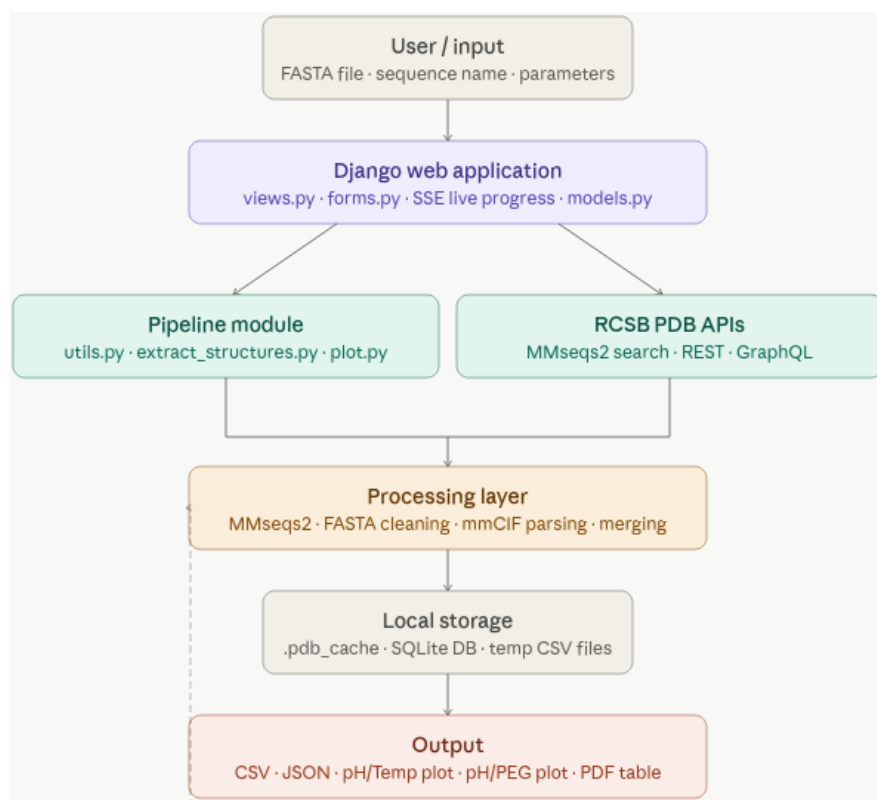


Figure 4.6: *Django app module*

Django serves as the central web framework that connects the user interface, backend pipeline, external APIs, and data processing components. In this project, Django is responsible for:

- **Handling user input**

Users upload the cleaned FASTA sequence and the name of the sequence type and configure the analysis parameters through Django forms ([forms.py](#)).

- **Managing application logic**

The [views.py](#) file is responsible for handling user requests, processing the sequence submitted, executing the bioinformatics pipeline, and returning the results to the web interface. It acts as the main controller between the frontend form and backend processing modules.

The file imports required Django modules, configuration settings, forms, utility

functions, and operating system utilities:

- `settings` are used to access the Django project configurations, such as the media directory.
- `render` generates HTML responses using templates.
- `SequenceForm` handles user input validation.
- The `run_pipeline` utility function is executed with the sequence data and the path of the media directory. This function performs sequence analysis and generates result files.

- **Integrating external services**

The `run_pipeline_view()` function processes both GET and POST requests from the user interface. Django communicates with the RCSB Protein Data Bank APIs (Search and GraphQL) to retrieve protein structure information. An empty dictionary named `context` is created to store data that will be passed to the HTML template.

- **Managing storage and results**

Processed data, cached PDB files, and analysis results are stored locally using SQLite and cache directories.

- **Delivering outputs to users**

When the pipeline executes successfully, the generated output files are extracted from the returned dictionary. The framework then provides downloadable results through the web interface, including CSV files, FASTA files, graphical plots, and PDF reports, as illustrated in Figure 4.7.

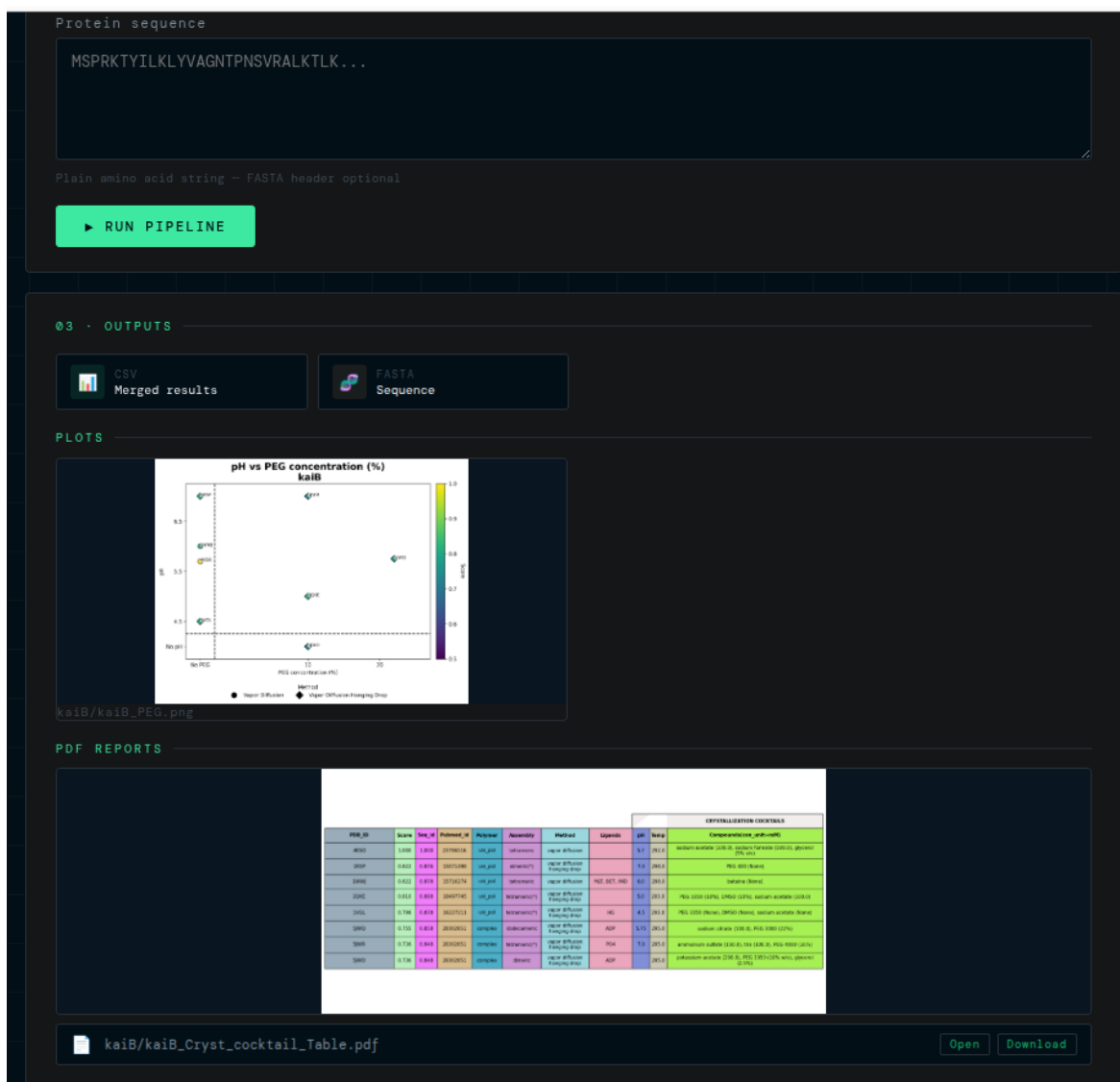


Figure 4.7: Result page of the webpage

In general, Django acts as the **orchestration layer** that integrates the frontend interface with bioinformatics processing tools and data resources.

4.5 FAIR Principles

The [FAIR Principles](#) provide guidance for scientific data management and are relevant to all stakeholders in the current digital world. Guidelines should be applied for the publication of digital resources such as datasets, code, workflows, and research objects, in a manner that makes them Findable, Accessible, Interoperable and Reusable (FAIR) [24, 33]. Although initially intended for a broad spectrum of digital assets, the FAIR Guiding Principles have been progressively adopted for research software, reflecting its growing importance in enabling transparent, reproducible, and reusable research. The FAIR for Research Software (FAIR4RS) Working Group (WG) has adapted the FAIR Guiding Principles to create the FAIR Principles for Research Software (FAIR4RS Principles) [34, 35, 36]. Research software is defined by the FAIR4RS WG as including

“source code files, algorithms, scripts, computational workflows and executables that were created during the research process or for research purpose. Software components (e.g., operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or with a clear research intent should be considered software in research and not Research Software[37].

Box | FAIR Principles for Research Software

F: Software, and its associated metadata, is easy for both humans and machines to find.

F1. Software is assigned a globally unique and persistent identifier.

F1.1. Components of the software representing levels of granularity are assigned distinct identifiers.

F1.2. Different versions of the software are assigned distinct identifiers.

F2. Software is described with rich metadata.

F3. Metadata clearly and explicitly include the identifier of the software they describe.

F4. Metadata are FAIR, searchable and indexable.

A: Software, and its metadata, is retrievable via standardised protocols.

A1. Software is retrievable by its identifier using a standardised communications protocol.

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 software is no longer available.

I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.

I2. Software includes qualified references to other objects.

R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).

R1. Software is described with a plurality of accurate and relevant attributes.

R1.1. Software is given a clear and accessible license.

R1.2. Software is associated with detailed provenance.

R2. Software includes qualified references to other software.

R3. Software meets domain-relevant community standards.

4.5.1 Implementation of FAIR for Data Management

The data extracted and produced by the pipeline originate from the Protein Data Bank which is a special purpose data archive that contains information about 3D structures of proteins and nucleic acids. Each entry is represented by a DOI (F, A). Other wwPDB access points provide access to wwPDB records through URLs that are likely to be stable in the long-term (F), and all data and metadata are searchable through one or more of the wwPDB-affiliated websites (F). All wwPDB entries are hosted on an FTP server (A) and are represented in machine readable formats (text and XML). Structural metadata are retrieved from mmCIF files, which is the standard data format of the International Union of Crystallography (IUCr) (F, I), also contain

cross-references to common identifiers such as PubMed and NCBI Taxonomy, and are described in data dictionaries and schema documents (<http://mmcif.wwpdb.org> and <http://pdbml.wwpdb.org>) which conform to the IUCr data standard for the chemical and structural biology domains (R). Various software tools are available to interpret both wwPDB data and metadata (I, R).

Additionally, The pipeline provides an automated FAIR-compliant metadata generator for the output files. Its aim is to ensure that every file produced by the crystallization data extraction process (CSVs, FASTA sequences, json, plots, and PDF reports) is automatically documented with rich and standardized metadata at the time of creation. By assigning unique identifiers, computing SHA-256 checksums for integrity verification, classifying files by type and purpose, and embedding provenance information (authorship, software version, data source, license) using DCAT and JSON-LD vocabulary, the generator ultimately supports transparency, reproducibility, and long-term reuse of the crystallization dataset by other researchers. Figure 4.9 below is an example of the `kaiB_metadata.json` file.

```
{
  "metadata_version": "1.0",
  "metadata_standard": "FAIR Principles Compliant",
  "generation_timestamp": "2026-06-11T13:08:58.869792Z",
  "generation_timestamp_human": "2026-06-11 13:08:58 UTC",
  "dataset": {
    "@context": "https://www.w3.org/ns/dcat",
    "@type": "dcat:Catalog",
    "dataset_id": "PCDE-20260611-150858-872458",
    "dataset_name": "PCDE Analysis Results - kaiB",
    "dataset_description": "Comprehensive protein crystallization data extraction results for sequence: kaiB",
    "creation_date": "2026-06-11T13:08:58.869792Z",
    "creation_date_human": "2026-06-11 13:08:58 UTC",
    "software": {
      "name": "Protein Crystallization Data Extraction (PCDE)",
      "version": "1.0.0",
      "release_date": "2026-01-01",
      "url": "https://github.com/RitAreaSciencePark/Protein_Crystallization_Data_Extraction"
    },
    "organization": {
      "id": "rit-area-science-park",
      "name": "RitAreaSciencePark",
      "url": "https://github.com/RitAreaSciencePark"
    },
    "author": {
      "name": "Nana Njantang Ruth",
      "orcid": "https://orcid.org/0000-0002-6003-7521"
    },
    "data_source": {
      "primary_database": "RCSB Protein Data Bank (PDB)",
      "database_url": "https://www.rcsb.org/",
      "search_scope": "Protein structures with X-ray crystallography experimental conditions",
      "api_endpoints": [
        "https://search.rcsb.org/rcsbsearch/v2/query",
        "https://data.rcsb.org/rest/v1/core/entry"
      ]
    },
    "metadata_standards": {
      "standards": [
        "FAIR",
        "Dublin Core",
        "RDF/JSON-LD"
      ],
      "version": "1.0",
      "compliance_level": "Full compliance with FAIR principles"
    },
    "access": {
      "url": "https://github.com/RitAreaSciencePark/Protein_Crystallization_Data_Extraction",
      "type": "public"
    }
  },
  "files": [
    {
      "@context": "https://www.w3.org/ns/dcat",
      "@type": "dcat:Dataset",
      "file_id": "kaiB_sequence-20260611-150858",
      "file_name": "kaiB_sequence.fasta",
      "file_path": "/home/ruth/Protein_Crystallization_Data_Extraction/output/kaiB/kaiB_sequence.fasta",
      "relative_path": "kaiB_sequence.fasta",
      "file_properties": {
        "size_kb": 0.11,
        "file_extension": ".fasta",
        "encoding": "utf-8"
      },
      "checksum": {
        "algorithm": "SHA-256",
        "value": "f24ab10c123b7f699fe1918f6a1cb9cef5f1da5bceddc700a987bd2b6f58476c"
      },
      "description": "Input sequence in FASTA format (kaiB_sequence.fasta)",
      "category": "fasta",
      "input_context": {
        "sequence_type": "kaiB"
      }
    }
  ]
}
```

Figure 4.8: *Generated Metadata for the output folder and output file for a kaiB protein*

4.5.2 FAIR for Research Software Assessment: PCDE Project

Software Findability

The software is available on GitHub at https://github.com/RitArea-SciencePark/Protein_Crystallization_Data_Extraction and has been assigned a unique identifier in Zenodo [F1]. Modules and functions are identifiable by name [F1.1]. Each release has a persistent Git tag, a release page, and downloadable archives (ZIP and TAR.GZ). For example, v1.0.0 corresponds to the initial release tag, while v1.0.1 corresponds to a bug-fix release [F1.2]. The software is provided with comprehensive metadata available in the `pcde_metadata.yaml` file that clearly and explicitly include the identifier of the software they describe (Release links point to specific tags (v1.0.0, v1.0.1) [F2].

Software Accessibility

The software is retrievable by its identifier using a standardized communication protocol, as it depends on github, which is a public repository (there is no authentication required for public repositories). The repository accessible via HTTPS and Git clone via: `git clone https://github.com/RitArea-SciencePark/Protein_Crystallization_Data_Extraction.git`. The zip archives which are directly downloadable links are included in the release notes [A1]. The HTTPS is universally implementable and works on any platform (linux, macOS, windows). The git protocol is free of use, universally available on all platforms, and widely adopted (industry standard). All dependencies are freely available and no commercial software or licenses are required for installation [A1.1]. Cloning the repository does not require credentials, which makes it appropriate for open science. The github supports SSH key authentication for pushing changes, personal access tokens (PAT) for programmatic access, and two factor authentication (2FA) for account security. The web application implemented in Django supports authentication out-of-the-box, and currently no login is required [A1.2].

Metadata is accessible even when the software is no longer available. The release pages remain accessible even if the code is archived. The Release notes document features, dependencies, and installation. The CITATION.cff remains accessible in commits, and the archives (ZIP, TAR.GZ) are available from the release pages [A2].

Software Interoperability

The software reads, writes, and exchanges data in a way that meets domain-relevant community standards [I1]:

- **Input Data Format:** accepts plain text sequences (amino acids, DNA, RNA) and FASTA format, which are the bioinformatics standard and ISO 81-compatible.
- **The output data files align with the community standard:** the CSV is Universal tabular format, compatible with R, Python, Excel, statistical software. The FASTA (input/output sequence format) is recognized by BLAST, MMseqs2, Clustal, alignment tools. PDF and PNG files are standard publication visualization formats. The JSON is a machine-readable metadata format.

- **Input Data Format aligns with standard protocols:** the HTTP REST API calls to RCSB PDB (standard HTTP). The mmCIF parsing via the gemmi library is in accordance with the international crystallography standard. Pandas dataFrames is the de facto standard for Python data science.

Software includes qualified references to other objects [I2]:

- **Internal Dependencies:** This modular structure ensures that the dependencies are clearly defined, facilitating the maintainability, readability, and traceability of the code.
- **External Software Dependencies:** All external software dependencies are listed in the `requirements.txt` file together with their corresponding version specifications. Dependencies are further documented in the README under the *Module dependencies* section, where the purpose of each module and its associated dependencies are clearly described. This ensures transparency, reproducibility, and ease of maintenance by providing a comprehensive overview of the software's dependency structure.
- **Data Provenance Links:** Every output record is traceable through its associated PDB_ID. Each PDB_ID can be resolved to the corresponding RCSB PDB web record through a persistent URL while each PubMed_ID can be resolved to the corresponding publication in the NCBI PubMed database through a persistent URL.

Software Reusability

The software is described with a plurality of accurate and relevant attributes [R1].

Documentation and Metadata Compliance: The project demonstrates comprehensive compliance with documentation and metadata best practices. Its documentation is extensive, comprising a `README.md` file covering the overview, key features, workflow, installation, usage, module reference, dependencies, releases, citation, and license. Additional documentation includes a `RELEASE_NOTES_v1.0.0.md` file, a `CITATION.cff`. The software attributes are comprehensively described and depend on Python 3.10+ and more than 35 external packages with specified versions, operate on Linux, macOS, and Windows platforms, and are implemented predominantly in Python (89.5% of the codebase). Quality attributes are also well defined, including data validation through X-ray crystallography filtering, error handling that excludes entries with missing critical parameters, performance optimization through `ThreadPoolExecutor` and caching, and reproducibility ensured by version-controlled source code and explicitly specified dependencies. Metadata attributes are structured and include author names, ORCID identifiers, organizational affiliation (RitAreaSciencePark), license information (MIT), version history (1.0.0, 1.0.1), release date (2026-06-09), repository URL and references to related publications and methodologies. The software supports deployment as a standalone application, web interface, or batch-processing workflow, while acknowledging known limitations such as dependence on compound annotation quality and the need for standardization of free-text descriptions.

The software is given a clear and accessible license MIT (© 2025 RitAreaSciencePark)

that Permits reuse, modification, distribution, and commercial use [R1.1].

Data Source Provenance. The pipeline ensures explicit data provenance by retrieving structural and experimental information exclusively from authoritative public repositories. The primary source is the RCSB Protein Data Bank (PDB), accessed through its official Search and Data REST APIs, which are fully documented to ensure transparency and reproducibility of data retrieval. Additional publication metadata are obtained from PubMed to provide experimental and bibliographic context. To facilitate complete traceability, every record generated by the pipeline retains the corresponding PDB_ID and PubMed_ID, allowing users to directly reference the original structural entry and its associated scientific publication.

Reproducibility. The pipeline was designed to ensure a high level of reproducibility through the use of a well-defined software environment and standardized execution procedures. Detailed setup instructions are provided for Linux, macOS, and Windows platforms, together with exact dependency versions and virtual environment configuration. The pipeline also generates a predictable output directory structure, with input FASTA sequences archived alongside all corresponding results to facilitate traceability, verification, and the reproducible execution of analyses.

Chapter 5

Conclusion

More than 90% of the structures available in the PDB have been determined using X-ray diffraction (XRD), an experimental technique that requires the sample to be in crystalline form. In this method, a macromolecular crystal is exposed to a X-ray beam, and the resulting diffraction pattern is used to determine its structure.

A critical step in this process is the production of suitable crystals, and identifying the conditions that promote crystal formation remains a central challenge in structural biology. Numerous studies have shown that macromolecular crystallization depends on a wide range of factors, including the identity and concentration of chemical components in the crystallization condition (cocktail), the pH of the sample and/or solution, incubation temperature, and other environmental parameters.

By leveraging the wealth of historical crystallization data available in the PDB archive, researchers can reduce the reliance on extensive trial and error screening and instead focus on optimizing conditions for high quality crystal growth. The primary goal of this project is to collect and analyze sufficient data to better understand the complex relationships between crystallization parameter space and successful crystal formation. This is achieved by extracting and characterizing the most prevalent chemical components and experimental conditions associated with successful macromolecular crystallization.

The wealth of information on crystallization conditions available in the PDB is often difficult to access in a standardized format. To address this, in this work we provide crystallization data extracted using custom Python scripts in combination with the PDB APIs. The first step involves querying structures with sequences similar to the input sequence, reporting the similarity score, entity number, sequence identity (based on the MMseqs2 alignment tool), and E-value. Subsequently, for each identified hit, the code developed retrieves metadata related to crystallization experiments from either the mmCIF files or the GraphQL interface of the PDB archive.

These metadata include sequence identity, score, `PubMed_ID`, resolution, polymer type, assembly, experimental method, `pdbx_ph_range`, ligands, and details of the crystallization cocktail, such as pH, temperature, and `pdbx_details`. The `pdbx_details` field is further parsed into a list of compounds with standardized chemical names and

concentration units, which are then appended to the extracted metadata. Finally, all results are consolidated into a single .csv file.

The extracted data are further processed by grouping PDB entries that share identical experimental conditions into a single record. The results are then presented in a visually intuitive, colored PDF format to facilitate interpretation. In addition, a scatter plot of pH versus PEG concentration is generated to explore potential trends between these variables. All data are made easily accessible through a dedicated webpage, “*Crystallization Cocktail*”. The cumulation these crystallization information allows the researchers to design crystallization experiments depending on the character of the macromolecules and also provides a foundation for further analysis of relationships that may contribute to optimizing macromolecular crystallization processes.

The main limitation of this project lies in the parsing of the free-text field (`pdbx_details`), which currently achieves an accuracy of approximately 80%. Future work will focus on improving this step, potentially by leveraging AI-based tools to more reliably extract chemical components and their corresponding concentrations from unstructured text and group them in classes of precipitants, buffers, additives, salts etc.

Appendix A

Multiple Sequence Fasta file Scripts

A.1 `scripts_fasta_file` Folder

The `scripts_fasta_file` folder contains the scripts responsible for extracting crystallization data from the PDB and exporting the results to CSV format. The main components of this pipeline are described below.

A.1.1 `main.py`

This script serves as the primary entry point of the pipeline. It imports all required functions from `extract_data_fasta.py` and orchestrates the full processing workflow.

`cleaning_and_read_fasta_file.py` This module reads a raw FASTA file, removes gap characters ("-") from sequences, and returns a dictionary mapping each sequence identifier to its corresponding cleaned amino acid sequence. It is designed to handle multi-sequence FASTA files.

`extract_data_fasta.py` The cleaned sequences generated in the previous step are used to perform sequence similarity searches against the RCSB PDB. The script queries the RCSB Search API using a JSON formatted request derived from each input sequence and filters the results to retain only structures determined by X-ray crystallography. To maximize retrieval coverage, a sequence identity threshold of 30% is applied.

For each retrieved PDB entry, the script extracts the PDB identifier and associated similarity score. The function `extract_crystallization(pdb_id)` retrieves experimental crystallization metadata from the corresponding mmCIF file. Parsing is performed using the `gemmi` library, which accesses the primary data block via `doc.sole_block()` and returns a structured dictionary of crystallization parameters.

The resulting CSV file includes the following fields for each entry: `query_ID`, `pdb_id`, `score`, and the associated experimental crystallization metadata.

A.1.2 `main2.py`

This script follows the same overall structure as `main.py`, importing functions from `cleaning_and_read_fasta_file.py` and the PDB processing module.

The key difference with respect to `main.py` is that the output CSV file contains aggregated crystallization data for *all* sequences in the input FASTA file. Duplicate PDB identifiers are removed so that each structure is represented only once across the full dataset.

Bibliography

- [1] J. S.-I. Kwon, M. Nayhouse, P. D. Christofides, and G. Orkoulas, “Modeling and control of protein crystal shape and size in batch crystallization,” *AIChE Journal* **59** no. 7, (2013) 2317–2327.
- [2] J. R. Luft, J. Newman, and E. H. Snell, “Crystallization screening: the influence of history on current practice,” *Structural Biology and Crystallization Communications* **70** no. 7, (2014) 835–853.
- [3] A. McPherson and J. A. Gavira, “Introduction to protein crystallization,” *Structural Biology and Crystallization Communications* **70** no. 1, (2014) 2–20.
- [4] J. Kirkwood, D. Hargreaves, S. O’Keefe, and J. Wilson, “Analysis of crystallization data in the protein data bank,” *Structural Biology and Crystallization Communications* **71** no. 10, (2015) 1228–1234.
- [5] C. Campos-Escamilla, L. A. Gonzalez-Ramirez, F. Otalora, J. A. Gavira, and A. Moreno, “A short overview on practical techniques for protein crystallization and a new approach using low intensity electromagnetic fields,” *Progress in Crystal Growth and Characterization of Materials* **68** no. 1, (2022) 100559.
- [6] M. L. Lynch, M. F. Dudek, and S. E. Bowman, “A searchable database of crystallization cocktails in the pdb: analyzing the chemical condition space,” *Patterns* **1** no. 4, (2020) 100024.
- [7] G. R. Budziszewski, V. Stojanoff, and S. E. J. Bowman, “Preparing for successful protein crystallization experiments,” *Acta Crystallographica Section F* **81** no. 7, (Jul, 2025) 272–280. <https://doi.org/10.1107/S2053230X25004650>.
- [8] T. M. Bergfors, *Protein crystallization*. Internat’l University Line, 2009.
- [9] M. L. Lynch, M. E. Snell, S. A. Potter, E. H. Snell, and S. E. Bowman, “20 years of crystal hits: progress and promise in ultrahigh-throughput crystallization screening,” *Biological Crystallography* **79** no. 3, (2023) 198–205.
- [10] N. Rosa, M. Ristic, L. Thorburn, G. J. Abrahams, B. Marshall, C. J. Watkins, A. Kruger, A. Khassapov, and J. Newman, “Tools to ease the choice and design of protein crystallisation experiments,” *Crystals* **10** no. 2, (2020) 95.
- [11] Y. Liu, H. Hou, J. Li, Q.-D. Cheng, X. Zhang, X.-B. Zeng, A. Fiaz, B. Wang,

- C.-Y. Zhang, Q.-Q. Lu, *et al.*, “Direct crystallization of proteins from impure sources,” *Crystal Growth & Design* **20** no. 3, (2020) 1694–1705.
- [12] G. E. Dale, C. Oefner, and A. D’Arcy, “The protein as a variable in protein crystallization,” *Journal of structural biology* **142** no. 1, (2003) 88–97.
- [13] N. E. Banayan, B. J. Loughlin, S. Singh, F. Forouhar, G. Lu, K.-H. Wong, M. Neky, H. S. Hunt, L. B. Bateman Jr, A. Tamez, *et al.*, “Systematic enhancement of protein crystallization efficiency by bulk lysine-to-arginine (kr) substitution,” *Protein Science* **33** no. 3, (2024) e4898.
- [14] Z. S. Derewenda and P. G. Vekilov, “Entropy and surface engineering in protein crystallization,” *Biological crystallography* **62** no. 1, (2006) 116–124.
- [15] A. McPherson and B. Cudney, “Optimization of crystallization conditions for biological macromolecules,” *Structural Biology and Crystallization Communications* **70** no. 11, (2014) 1445–1467.
- [16] I. Dinç, M. L. Pusey, and R. S. Aygün, “Optimizing associative experimental design for protein crystallization screening,” *IEEE transactions on nanobioscience* **15** no. 2, (2016) 101–112.
- [17] M. J. Cox and P. C. Weber, “An investigation of protein crystallization parameters using successive automated grid searches (sags),” *Journal of Crystal Growth* **90** no. 1-3, (1988) 318–324.
- [18] R. Giegé, “A historical perspective on protein crystallization from 1840 to the present day,” *The FEBS journal* **280** no. 24, (2013) 6456–6497.
- [19] A. McPherson and B. Cudney, “Searching for silver bullets: an alternative strategy for crystallizing macromolecules,” *Journal of structural biology* **156** no. 3, (2006) 387–406.
- [20] S. Larson, J. Day, C. Nguyen, R. Cudney, and A. McPherson, “Progress in the development of an alternative approach to macromolecular crystallization,” *Crystal Growth and Design* **8** no. 8, (2008) 3038–3052.
- [21] P. D. Bank, “Protein data bank,” *Nature New Biol* **233** no. 223, (1971) 10–1038.
- [22] wwPDB consortium, “Protein data bank: the single global archive for 3d macromolecular structure data,” *Nucleic acids research* **47** no. D1, (2019) D520–D528.
- [23] wwPDB consortium, “Protein data bank: the single global archive for 3d macromolecular structure data,” *Nucleic Acids Research* **47** no. D1, (2019) D520–D528.
- [24] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, *et al.*, “The fair guiding principles for scientific data management and stewardship,” *Scientific data* **3** no. 1, (2016) 1–9.

- [25] J. Y. Young, J. D. Westbrook, Z. Feng, R. Sala, E. Peisach, T. J. Oldfield, S. Sen, A. Gutmanas, D. R. Armstrong, J. M. Berrisford, *et al.*, “Onedep: unified wwPDB system for deposition, biocuration, and validation of macromolecular structures in the PDB archive,” *Structure* **25** no. 3, (2017) 536–545.
- [26] J. Westbrook, H. Berman, and S. Hall, “International tables for crystallography, vol. g,” *Definition and Exchange of Crystallographic Data* (2005) 61–70.
- [27] J. Westbrook, N. Ito, H. Nakamura, K. Henrick, and H. M. Berman, “Pdbml: the representation of archival macromolecular structure data in xml,” *Bioinformatics* **21** no. 7, (2005) 988–992.
- [28] A. R. Kinjo, H. Suzuki, R. Yamashita, Y. Ikegawa, T. Kudou, R. Igarashi, Y. Kengaku, H. Cho, D. M. Standley, A. Nakagawa, *et al.*, “Protein data bank japan (pdbj): maintaining a structural data archive and resource description framework format,” *Nucleic acids research* **40** no. D1, (2012) D453–D460.
- [29] S. F. Altschull, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” *J. mol. Biol* **215** no. 3, (1990) 403–410.
- [30] E. S. Donkor, N. Dayie, T. K. Adiku, *et al.*, “Bioinformatics with basic local alignment search tool (blast) and fast alignment (fasta),” *Journal of Bioinformatics and sequence analysis* **6** no. 1, (2014) 1–6.
- [31] M. Steinegger, M. Mirdita, E. L. Karin, L. Von Den Driesch, C. Galiez, and J. Söding, “Mmseqs2 user guide,” 2019.
- [32] M. Steinegger and J. Söding, “Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets,” *Nature biotechnology* **35** no. 11, (2017) 1026–1028.
- [33] M. D. Wilkinson, S.-A. Sansone, E. Schultes, P. Doorn, L. O. Bonino da Silva Santos, and M. Dumontier, “A design framework and exemplar metrics for fairness,” *Scientific data* **5** no. 1, (2018) 180118.
- [34] A. Struck, “Research software discovery: an overview,” in *2018 IEEE 14th international conference on e-Science (e-Science)*, pp. 33–37, IEEE. 2018.
- [35] H. Shanahan, “Fair principles for research software (fair4rs principles),”.
- [36] M. Barker, N. P. Chue Hong, D. S. Katz, A.-L. Lamprecht, C. Martinez-Ortiz, F. Psomopoulos, J. Harrow, L. J. Castro, M. Gruenpeter, P. A. Martinez, *et al.*, “Introducing the fair principles for research software,” *Scientific data* **9** no. 1, (2022) 622.
- [37] M. Gruenpeter, D. S. Katz, A.-L. Lamprecht, T. Honeyman, D. Garijo, A. Struck, A. Niehues, P. A. Martinez, L. J. Castro, T. Rabemanantsoa, *et al.*, “Defining research software: a controversial discussion,” *Zenodo* (2021) .