Newcastle University

School of Computing

Computational Approaches to Generating Diverse Enzyme Panels

Christian J.I. Atallah

May 2022

Abstract

Motivation

Enzymes are complex macromolecules crucial to life on earth. From bacteria to human beings, all organisms use enzymes to catalyse the many thousands of chemical reactions occurring in their cells. Enzyme functions are so diverse that the use of enzymes in industries like pharmaceuticals and agriculture has gained popularity over recent years as ”biocatalysts”.

Unfortunately, the confident laboratory-based characterisation of enzyme function has lagged behind a massive increase in sequencing data, slowing down initiatives that look to use biocatalysts as part of their chemical processes. Computational methods for identifying biocatalysts do exist, but often falter due to the complexity of enzymes and sequence bias, leaving much of the catalytic space of enzymes and their families undiscovered.

This thesis has two major themes: the development of *in silico* approaches for curating diverse panels of novel enzyme sequences for experimental characterisation, and of tooling that integrates *in silico* panel creation and *in vitro* enzyme characterisation into a unified and iterative framework.

Contributions of this thesis

The contributions of this thesis can be divided into the two larger themes, starting with the diverse panel selection of sequences from an enzyme family:

- A novel type of protein network based on patterns of coevolving residues that can be used to identify functionally-interesting groupings in enzyme families.
- The automatic sampling of functionally diverse subsets of enzyme sequences by solving the maximum diversity problem.

- A study into the viability of artificially increasing enzyme family diversity through neural networks-based generation of synthetic sequences.

The second theme, which deals with built tools for bridging the gap between the *in silico* and *in vitro* side of enzyme family exploration:

- A platform that integrates the panel selection process and resulting characterisation data to promote an iterative approach to exploring enzyme families.
- A repository for storing the metadata generated by the major steps of characterisation assays in the lab.

Declaration

I declare that this thesis is my own work unless otherwise stated. No part of this thesis has previously been submitted for a degree or any other qualification at Newcastle University or any other institution.

Christian J.I. Atallah

May 2022

Publications

Portions of the work within this thesis have been documented in the following publications:

Atallah, C., Skelton, D.J., Charnock, S.J. and Wipat, A., 2019. Functional Analysis of Enzyme Families Using Residue-Residue Coevolution Similarity Networks. bioRxiv, p.646539.

McLaughlin, J.A., Myers, C.J., Zundel, Z., Wilkinson, N., **Atallah, C.** and Wipat, A., 2018. sboljs: Bringing the synthetic biology open language to the web browser. ACS synthetic biology, 8(1), pp.191-193.

Acknowledgements

I would first like to thank my two supervisors, Professor Anil Wipat and Professor Simon Charnock, for the massive amounts of support they provided to me over the last four years. I will forever be thankful to you for giving me this opportunity, and for all of the doors that have opened to me as a result. I would also like to acknowledge the EPSRC and Prozomix Limited for the funding that made this PhD possible. Thank you as well to Dr. James Finnigan for help with experimental work.

A huge thanks also goes out to all of the great members of the IntBio team and others who have made the last four years so enjoyable. I was extremely lucky to spend so much time with as many fun and welcoming people. In particular, I would like to thank James^2 (real and fake), Bill, David, Bradley, and Jasmine, not just for the memories, but also for all of the help provided during my thesis. Whether you helped debug code, gave feedback on the write-up, or accompanied me on Shijo trips, I am grateful.

There are many more friends I would like to thank, old and new, close and far: Polly, Ruud, Carol, Elisa, Leen, Mahra, Dina, Yamane, Liza, Rana, Dani, Patrick, Jake, and the Space Vandals. All of you have had an important part to play in making this thesis possible, and I am thankful for the great company over the years.

I would like to dedicate this thesis to my wonderful parents, Joseph and Isa. Everything I have is only possible thanks to the unending support, belief, and love you've given me all of my life. Words cannot explain well enough just how eternally thankful I am to you. I would also like to dedicate this thesis to loved ones who are not here anymore: Micha, Lily, Kenzo, I miss you every day. I would also like to thank my two siblings, Nathalie and Antoine, for always believing in me.

And last, but most definitely not least, I would like to thank my amazing partner and my rock, Nicole. You kept me grounded and sane when things got difficult, and were the reason for so many amazing moments over the last few years. Your endless support was indispensable to me writing this thesis - I am forever lucky and grateful for having you in my life, not just for this adventure, but also for what comes next.

Contents

List of Figures

List of Tables

Acronyms

API application programming interface

DBMS database management system

SSN sequence similarity network

CSN coevolution similarity network

MDP maximum diversity problem

ECP equivalent coevolving pair

MSA multiple sequence alignment

BLAST Basic Local Alignment Search Tool

EC enzyme commission

DCA direct coupling analysis

WAMCC weighted-average MaxClust coverage

PDB Protein Data Bank

NP-hard non-deterministic polynomial-time hardness

IP InterPro

GSI Gini-Simpson index

HMM hidden markov model

GAN generative adversarial network

SPARQL SPARQL Protocol and RDF Query Language

AKR aldo-keto reductase

PCA principal component analysis

t-SNE t-distributed stochastic neighbour embedding

SDS-PAGE sodium dodecyl sulphate–polyacrylamide gel electrophoresis

I-TASSER iterative threading assembly refinement

NAD nicotinamide adenine dinucleotide

NADP nicotinamide adenine dinucleotide phosphate

TC total cell fraction

CFE cell free extract

SPR simple perfect repeat

GPU graphics processing unit

IntEnz-Lab Integrative Enzyme Lab

ChEBI Chemical Entities of Biological Interest

SBOL Synthetic Biology Open Language

DBTL design-build-test-learn

NGS next-generation sequencing

ENA European Nucleotide Archive

IRED imine reductases

NCBI National Center for Biotechnology Information

nr non-redundant

ORF open reading frame

ANN artificial neural network

iGEM international Genetically Engineered Machine

SEP SBOL enhancement protocol

LCR low complexity region

KEGG Kyoto encyclopedia of genes and genomes

IPTG isopropylthio-β-galactoside

CSV comma-separated values

RDF resource description framework

GAN generative adversarial network

GFP Green Fluorescent Protein

1

Introduction

Contents

1.1 Motivation for this research

Enzymes are complex macromolecules crucial to life on earth. From bacteria to human beings, all organisms use enzymes to catalyse the many thousands of chemical reactions occurring in their cells. Mostly consisting of proteins, the number of sequenced enzymes has increased dramatically thanks to the advent of next-generation sequencing.

Enzyme function is so diverse that the use of enzymes in industries like pharmaceuticals and agriculture has gained popularity over recent years [5]. Mostly, they are used to speed up synthesis reactions as ”biocatalysts” by multiple factors, and can sometimes even increase chemical yield by significant amounts. Also, biocatalysts are considered a type of ”green chemistry” as they categorically eliminate the need for environmentally harmful solvents and waste products.

Unfortunately, the confident laboratory-based characterisation of enzyme function and physiochemical properties - which is necessary to increase the portfolio of biocatalysts available to be used by industry - has lagged behind the massive increase in sequencing data, slowing down initiatives that look to use enzymes as part of their chemical processes. There is also a strong bias in the profile of enzymes that do get characterised, leaving much of the catalytic space of enzyme families undiscovered. Computational methods that attempt to functionally annotate enzymes do exist but often falter due to this bias and the complexity of enzyme homology assignment, resulting in unknown annotation at best, and incorrect annotation at worst on public databases.

Therefore, methods that specifically focus on speeding up and optimising the selection process of potentially novel enzyme sequences for characterisation would therefore have high impact, and is the major theme explored in this thesis. Finally, this work also tackles tooling that integrates both the in-silico and in-vitro sides of enzyme characterisation.

1.2 Research question

> “Can the diversity of enzyme panels curated from enzyme families for laboratory-based characterisation be further optimised using *in silico* approaches and tools?”

1.3 Aims and objectives

This research aimed to achieve the following two goals:

1. The development new computational methods that will help build diverse enzyme panels from enzyme families
2. The development of software tools that will help promote an integrated and iterative framework for the laboratory-based characterisation of enzyme families

These two research aims were explored through the following objectives:

- The development of novel enzyme similarity networks that are sensitive to detecting functional groupings at lower levels of sequence identity
- The development of algorithms for the automatic selection of diverse enzyme panels from larger enzyme family datasets
- The use of a sequence autoencoder to produce artificial enzyme diversity by generating synthetic enzyme sequences
- The development of a platform for integrating the process of in-silico sequence selection methods and enzyme characterisation assays
- The development of a repository for the storage of metadata arising from enzyme characterisation assays

1.4 Contributions of this research

The contributions of this research can be divided into two larger themes, starting with the selection of diverse enzyme sequences from an enzyme family:

- A novel type of similarity network based on patterns of coevolving amino acid residues, called Coevolution Similarity Networks (CSN). These networks can be used to analyse enzyme families and identify functionally-interesting clusterings
- Two implementations of algorithms that solve the maximum diversity problem (MDP) for the functionally diverse selection of enzyme sequences from a larger set
- An autoencoder neural network with a model refinement pipeline that can generate viable synthetic sequences of an enzyme family from a template set

The second theme, which deals with built tools for bridging the gap between the in-silico and in-vitro side of enzyme family exploration:

- IntEnz-Lab, which is a web-interface and repository that integrates the sequence selection process and experimental characterisation data to promote an iterative approach to discovering diversity in an enzyme family
- SynBioHub-Lab, which is a web-interface and repository for storing the metadata generated by the three major steps of the synthetic biology lifecycle with regards to enzyme characterisation, such as plasmid designs, laboratory protocols, and so on.

1.5 Thesis structure

The rest of this thesis is divided into six further chapters. Chapter 2 describes background research, the motivation behind this work, and a literature review of current approaches for functional characterisation of enzyme families.

The next four chapters are research chapters, and comprise the following:

- Chapter 3 discusses the development of CSNs for the functional analysis of enzyme families, and a comparative study with Sequence Similarity Networks (SSN).
- Chapter 4 explores the solving of the MDP for optimised and automatic selection of functionally diverse subsets from enzyme families, with two different algorithms implemented and compared to standard clustering methods.
- Chapter 5 described the development of a neural network-based method that generates synthetic sequences from a template set belonging to an enzyme family, and the likely structural and functional viability of such artificial sequences.
- Chapter 6 introduces two novel tools - IntEnz-Lab and SynBioHub-Lab - for a proof-of-concept approach to the exploration of enzyme families that is iterative and laboratory-based, which attempt to bridge the gap between the in-silico side and the experimental side of enzyme family analyses.

Finally, Chapter 7 concludes the thesis with a discussion that places the novel research in context with its motivations, and explores avenues for future work.

2

Background

Contents

2.1 Introduction

With the release of the first next-generation sequencing (NGS) technology in 2005 [6], the throughput of genome sequencing exploded. Ten years later, the price barrier of $1000 for sequencing a single human genome was breached [7], and as of August 2020 the National Human Genome Research Institute reported the average price of sequencing a human-sized genome has dropped to $689 [8]. It is now even possible to sequence the genomes of entire microbial communities, resulting in what are called 'metagenomes', which reveal the genomic content of all organisms existing in some sample, from the human gut flora [9] to deep-sea vent ecosystems [10].

This significant increase in the accessibility of high-throughput sequencing methods has similarly led to a boom in the number of genomic and proteomic sequences available in public databases. The European Nucleotide Archive (ENA) [11], which is a public database storing sequencing data since 1983, received in 2021 "over 700 000 submissions to the ENA in 12 months, comprising of 6600 studies, over a million samples and runs, and 160,000 (meta)genome assemblies" [12]. UniProt, which is a knowledgebase containing translated protein sequences from the ENA and other nucleotide sources, contained 45,288,084 sequence entries in the October 2013 release across thousands of different organisms [13]. However, this incredible number more than quadrupled in just under seven years, to 189,525,031 sequence entries in the April 2020 release [1].

Out of these rich and diverse resources of natural biological data, applications to various different fields were born. For example, enzymes are a class of organic macromolecules that organisms have evolved to use as catalysts for the countless chemical reactions ongoing in cells. As they are mostly proteins, the breadth of revealed enzyme sequences has also increased in magnitude. Between their naturally-evolved diversity and their optimisation as catalysts, applications involving enzymes as 'biocatalysts' have grown popular in various industries [14].

This chapter first consists of an review of biocatalysts, their applications, the current approaches employed in the generation of diverse enzyme panels, along with the limitations of such approaches, which is the principle theme of this thesis. Then, a literature review of three research branches provides necessary background knowledge relevant to this research; the different bioinformatics resources and techniques employed, the principles of machine learning and heuristic optimisation, and the application of synthetic biology in enzymology.

2.2 Enzymes as biocatalysts

Enzymes are organic macromolecules that catalyse chemical reactions in all known organisms. Mostly made up of protein members (with some being RNA-based), they are an essential component of complex life due to their incredible power at shortening the time chemical reactions take. For example, the decarboxylation of orotidine 5'-phosphate, which is an essential step of the synthesis of nucleic acids, is estimated to take 78 **million** years without its respective enzyme, which increases the rate of reaction by a factor of 10^{17} [15]. *Escherichia coli*, a key model prokaryotic organism, is known to have as many as 607 enzymes catalysing 744 different reactions organised into 131 different chemical pathways [16].

It is therefore clear that enzymes are not only essential to life on earth, but also have strong potential as tools to facilitate many different chemical processes used in modern industry. Owing to their diverse range of functions and immense efficiency as catalysts, various sectors of industry have been using enzymes as 'biocatalysts', a trend that is growing in popularity.

2.2.1 Applications of biocatalysts

Many different industries have recently adopted the use of enzymes as catalysts in synthesis reactions, with four main benefits to standard reactions (Figure 2.1):

1. High stereoselectivity and regioselectivity of reactions catalysed by enzymes due to enzyme specificity.
2. The simplification of chemical pathways through the circumvention of steps via enzyme catalysis.
3. The elimination of many chemical reagents and waste products necessary for standard synthesis workflows promoting a more 'green' chemistry.

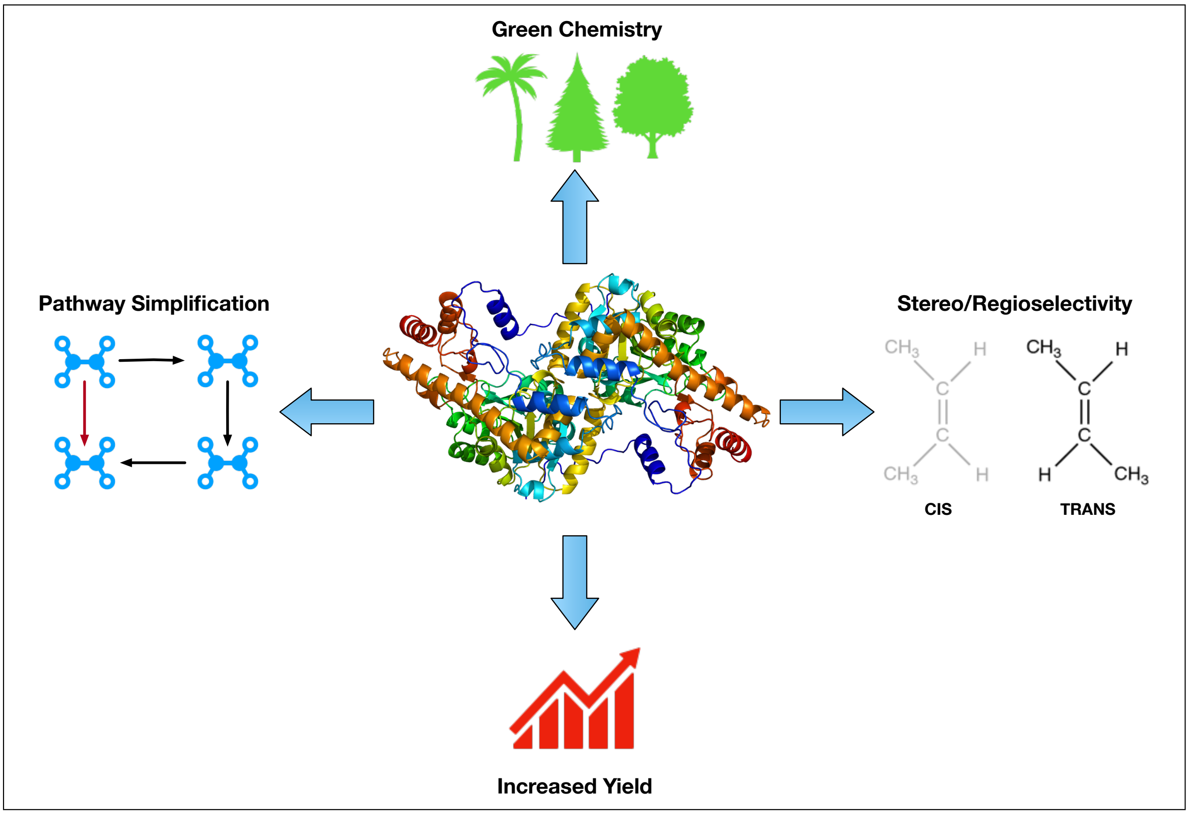


Figure 2.1: The use of enzyme biocatalysts has four main benefits compared to classic chemical synthesis; high stereo/regioselectivity, simplified pathways, a greener chemistry, and increased yield.

4. Increased chemical yield owing to the level of optimisation that enzymes have for their respective reactions.

For example, biocatalysts are used by the pharmaceutical industry in compound synthesis reactions, owing to their ”exquisite regioselective and stereoselective properties” that allow to bypass more complicated synthesis reactions [17]. Another benefit of using enzymes in pharmaceutical processes is the elimination of steps of a synthesis workflow, resulting in a reduction in time and costs . For example, transaminases are known to simplify such reactions by being able to directly convert ”ketones to chiral amines thereby circumventing multiple synthetic steps” [18].

Also, chemical processes often require reagents and produce waste products that are hazardous to the environment. The use of biocatalysts for many synthesis reactions removes the need for such reagents, making industrial processes more environmentally-friendly, and is often called ’green chemistry’ as a result [19]. For example, a ketone-reducing enzyme has been used in the synthesis of talampanol, which is a drug used for treating epilepsy and neurogenerative diseases. Per ton of talampanol produced, the use of a biocatalyst not only removed the need for over 300 thousand litres of solvents and three tons of chromium oxide (a suspected carcinogen), but it also increased the yield of the reaction from 16% standard chemical synthesis to 51% [19]. The yield was similarly doubled when a lipase was introduced for the production of a similar drug, Pregabalin, from 21% yield to 40% [19]. Such increases in yield, combined with the benefits to the environment, high stereo/regioselectivity, and the simplification of multi-step processes, shows the multi-faceted benefits of using enzymes in pharmaceutical manufacturing of drugs.

Similarly, biocatalysts have seen use in polymerisation reactions [19–22]. Just as with the synthesis of pharmaceutical agents, enzymes can help reduce the environmental impact of generating polymers. For example, polyaniline was successfully synthesised using a laccase [22]. Also, the use of enzymes in polymerisation reactions helps not just with regioselectivity, but also by simplifying the processes themselves, such as lipase-powered polymerisation of trimethylene carbonate which can be ”prepared in one-pot without the need of protection and deprotection chemistry” [19].

Other industries and applications where biocatalysts have proven useful include biofuels [23], waste-water treament [24], and the food industry [25], among others. Given the specificity of enzymes, and the far-reaching applications of them as biocatalysts, there is therefore a need for the identification and characterisation of more novel enzymes with the range of physiochemical and catalytic properties necessary to meet modern industrial processes.

2.2.2 Current in silico enzyme panel generation strategies

There is already a breadth of available sequences on public databases thanks to NGS methods, including metageomic sequencing. The development of NGS techniques has made it possible to not only explore the genome of an individual organism, but of entire communities in the form of metagenomes [26]. Metagenomes and the public databases they help populate are therefore hotbeds of enzymatic diversity which can be exploited for the purposes of industry.

However, the vast majority of enzymes in public databases are uncharacterised [27], complicating the process of choosing a biocatalyst that is useful to some industrial process of interest. Therefore, a major requirement for the increased use of biocatalysts is further development of the repertoire of available enzymes that have been confidently characterised.

The profiling of an enzyme's function and its physiochemical profile however is highly complex to perform *in silico*; modern function prediction tools like DEEPre [28] can only confirm enzyme function at a general level, with confident substrate-specificity predictions being out of reach. Therefore, enzyme function can currently only be confirmed at a high confidence level in the laboratory, usually through chemical assays on overexpressed enzyme samples. Consequently, the initiative to discover and characterise further potential biocatalysts faces a bottleneck in time, accessibility, and cost, as it is not trivial to test an enzyme for some activity.

There have been calls as far back as 2004 for collaborative efforts in the characterisation of unknown protein (and therefore enzyme) space, due to already existing concerns that such unannotated space "highlight(s) just one portion of our ignorance about the information content of genomes and our lack of fundamental knowledge about the function of so many of the building blocks of cells" [29]. However, NGS methods have only further worsened the concerns Roberts had in 2004.

Nonetheless, while the proportion of uncharacterised protein space has undoubtedly increased, as has the amount of 'discoverable' diversity. It is estimated that the number of discoverable and useful biocatalysts in metagenomic samples ranges from 1.4 to 19 per million base pairs [30]. For example, 4874 glycosyl hydrolase homologues were identified *in silico* in 46 metagenomes [31]. Given the high number of enzyme sequences that is ever growing, it has become more possible to mine putative enzymes of a sought-after function to generate panels of enzymes which can then be characterised in the laboratory. These panels, when designed intelligently, should carry enough sequence diversity to increase the chances of discovering novel enzymes that can catalyse one or more reactions of interest.

There are multiple approaches to curating such enzyme panels, but essentially all of them require an initial an dataset building step, whereby a large data resource like

UniProt or metagenomic data is mined for enzymes of some interest. For example, Velikogne and colleagues [32] looked to characterise imine reductases (IRED), which they did by using four known IREDs from literature as template sequences. These sequences were used to query UniProt Basic Local Alignment Search Tool (BLAST), the algorithm of which is described in section 2.3.3. The result of such queries is a list of sequences similar enough to the template IREDs, which were then manually filtered based on the expectation of certain conserved domains and residues. This bioinformatics approach resulted in 182 hits after filtering, a number too high to be characterised fully in the laboratory. Velikogne and colleagues therefore performed a phylogenetic analysis, producing a tree displaying the evolutionary relationships between all the hits and the template IREDs. Finally, they curated a panel of novel IREDs to test in the lab by randomly picking 15 sequences from the main sub-branches of the tree, 10 of which were found to indeed be IREDs in the laboratory.

Similarly, Bastard and colleagues [27] looked to characterise one of the many enzyme families of currently-unknown function curated by the Pfam database [33], DUF849. Starting with a set of 725 sequences and one template enzyme, a complex bioinformatics analysis comprising sequence similarity, phylogenetics, genomic-context clustering, and active site modelling, was used to filter the set down to a panel of 322 candidate sequences. Of these sequences, 124 could be overexpressed and characterised. Impressively, after testing these sequences on 17 substrates, 80 of the 124 enzymes showed activity against at least one substrate.

Vanaceck and colleagues [34] queried another large public database using BLAST: the non-redundant (nr) database provided by the National Center for Biotechnology Information (NCBI) [35]. Specifically, nr was mined for novel dehalogenases, followed by a hierarchical clustering [36] of the resulting 5661 hits . Based on this clustering, 953 enzymes were kept, which were themselves filtered down to a set of 658 hits after the removal of incomplete and degenerate sequences. However, Vanaceck and colleagues did not have the capabilities to characterise such a large set, and instead filtered out a panel of 20 diverse enzymes. Specifically, they manually picked 20 enzymes based on diverse complex factors, such as taxonomy, the predicted volume of the active site, predictions of solubility, and so on. Twelve of the enzymes in the panel were successfully characterised, with nine of them exhibiting dehalogenase activity.

In another example, Kim and colleagues [37] used keywords to retrieve 86 different glucosidases from the database GenBank [38] in their search for an enzyme for activity on the plant extract indican. A phylogenetic tree was constructed for these 86 sequences, which separated them into 11 subfamilies. One representative sequence was arbitrarily picked for each subfamily to create a panel of representative enzymes. These enzymes were then used as BLAST queries to NCBI, and eight hits were arbitrarily chosen for characterisation based on homology to the template sequences, one of which was highly active towards indican.

Finally, Baud and colleagues [39] looked to discover novel transaminases from newly-generated metagenomic data. Tongue scrapings from nine individuals were sampled, sequenced, and assembled in their study. The open reading frames (ORFs) contained were then identified and translated. These ORFs were then mined for putative transamianses using a hidden markov model (HMM) [33], another method of searching for novel sequences described further in section 2.3.3. The resulting panel was small, with just 15 novel sequences identified, 11 sequences characterised, and just 3 showing significant activity, though these hits are just for one metagenome combined from nine samples.

In all the examples discussed here, a common approach is to first mine a large sequence resource, which is usually either a public database [32, 34, 37], or an assembled metagenome for which the ORFs were retrieved [39]. The mining is performed using some form of common bioinformatics algorithm that can search for hits based on some template query, with the most popular method being BLAST. Then, depending on the number of hits and on the characterisation capacity available, the hits often need to be filtered down to fit said capacity. In these cases, a popular approach is to perform a phylogenetic analysis followed by sampling from branches interpreted as subfamilies. In some others, a more complex and manual analysis of the diversity existing in the mined hits is performed to try and maximise the number of catalytic functions in the panel.

2.2.3 Limitations of current enzyme panel generation approaches

The implication resulting from the approaches discussed in section 2.2.2 is that there is not yet a defined and consistent framework for the discovery and characterisation of novel enzymes that is optimised in simplicity, accessibility, scalability, and applicability. In particular, the current approaches to the filtering of hits that is often necessary due to the lab bottleneck have various limitations, which are discussed in this section.

First, while the mining of hits using BLAST [40] and other sequence-identity based methods remains powerful even three decades since its first implementation [41], the selection of diverse enzyme panels using such methods comes with key assumptions about the level of sequence identity needed to represent homology between enzymes. Indeed, multiple of the examples discussed in section 2.2.2 use sequence-based methods like phylogenetics or hierarchical clustering to group up similar or homologous enzymes, which are then sampled from to create a smaller list of candidate enzymes to characterise.

However, while it is accepted that a sequence identity threshold above 40% is high enough for two proteins to be homologous [42], literature shows that enzymes are more complex. In 2002, Rost noted that sequence bias in public databases led to an underestimation of how similar two enzyme sequences need to be to share the same exact function, with e-values of BLAST as significant as 10^{-50} still leading to errors in function assignment [43]. In 2003, Tian and Skolnick showed that a sequence identity threshold of 60% is necessary to transfer function at an accuracy of 90% [44].

Such high levels of sequence identity being necessary for exact function conservation implies that using sequence-identity methods to discern homologous enzymes for panel selection requires the use of more stringent thresholds to be more precise. However, this process is further complicated by our current understanding of the evolution of enzyme functional diversity. For example, many protein family classifications are based on arbitrary sequence similarity thresholds [45], which ignores functional similarities between enzymes of low sequence similarity. For example, lactonases exhibiting phosphotriesterase activity have been found in three different superfamilies in an example of 'convergent evolution' [46], with further examples found in the enolase superfamily [47]. Also, new research seems to indicate that promiscuous enzymes, which catalyse more than one reaction, are more common than once thought, which further diversifies enzyme functionality beyond a sequence similarity lens [46–48].

There is therefore a need for panel selection methods that are based on more than sequence identity-based homology, and that are more specialised towards the features of enzymes like active sites [31]. Indeed, many of the approaches presented in section 2.2.2 use further sequence analysis methods as filtering steps, either to remove enzymes unlikely to be of the family of interest [32, 34], or to filter the amount of hits down to panels of a more convenient size for characterisation [27]. Unsurprisingly, these approaches were the most successful in terms of the proportion of panel enzymes that were active, at success rates of 66%, 64.5%, and 75%, for Velikogne and colleagues [32], Vanaceck and colleagues [34], and Bastard and colleagues [27], respectively. The works of Kim and colleagues [37] and Baud and colleagues [39] however performed no such analyses, leading to lower enzyme success rates of 12.5% for the former and 27.2% for the latter.

However, the more successful studies require a large amount of knowledge about the enzyme family of interest to perform the analysis. The work of Bastard and colleagues [27] would not have been possible without the previous resolution of the tertiary structure of one of the DUF849 enzymes. Velikogne and colleagues would not have been able to perform filtering of the hits based on the active site conservation without the already existing tertiary structure of one of the used template IREDs [32], similarly to Vanaceck and colleagues [34]. This requirement for expert knowledge of an enzyme family of interest is a significant bottleneck given the number of enzyme families of completely unknown function was as high as 22% in 2013 [27].

Also, having such knowledge about an enzyme family does not trivialise the process of panel generation by any means. For example, an enzyme labelled to be part of the strictosidine synthases subgroup of the nucleophilic attack, 6-bladed-propeller superfamily, with a mostly conserved active site and known to be "among the most similar to the experimentally characterized strictosidine synthases was shown to have ... no detectable strictosidine synthase activity" [47]. Indeed, erroneous annotations on public databases are common; it was estimated that two million proteins were given incorrect taxonomy annotation on the nr database, for example. Inconsistent annotation also exists between different databases, with disagreements between UniProt and the Kyoto encyclopedia of genes and genomes (KEGG) being as high 31% of annotated enzymes, ”showing that the two knowledge bases curators have different scientific opinions in many cases” [49] There is also a large amount of bias in the types of catalytic functions that are studied. For example, it was found in 2012 that ”80% of [enzyme] classes annotate only about 10% of UniProt enzymes, while the remaining 20% most common [enzyme] classes annotate 90% of UniProt enzymes” [49]. Therefore, even existing annotation can be unreliable without experimental evidence, further limiting how much existing knowledge is useful for making decisions about enzyme panel selection.

Furthermore, many of the in-depth analyses performed by the more successful enzyme panel studies can be difficult and time-consuming. In both the works of Bastard and colleagues and Vanaceck and colleagues a considerable level of manual analysis and interpretation was performed, using sequence similarity, phylogeny, active site models, solubility predictions, genomic-context clustering, and so on [27, 34]. The level of confident knowledge necessary to better sample new potential diversity in unknown sequence space is therefore high while also being hard and time-consuming to interpret.

Therefore, three main limitations in current approaches for generating diverse enzyme panels were identified and addressed in this research:

1. An over-reliance on raw sequence-identity based methods of grouping functionally similar enzymes
2. A large burden of knowledge necessary about enzyme families of interest to perform more in-depth analyses of existing diversity
3. A manual interpretation of such analyses that is complex and time-consuming that is required to properly sample diverse enzyme panels

In this thesis, novel computational methods that tackle these three limitations were therefore developed, specifically in Chapters 3, 4, and 5.

2.3 Bioinformatics for enzyme research

Bioinformatics is a field of research with the following definition:

> “
>
> Bioinformatics is conceptualizing biology in terms of macromolecules and then applying “informatics” techniques to understand and organize the information associated with these molecules, on a large-scale [50].
>
> ”

Enzyme research employs a breadth of different bioinformatics approaches. Indeed, much of the research performed in this work employs numerous bioinformatics tools and resources useful for enzyme research. In this section, a background of the bioinformatics relevant to this thesis is described in depth.

2.3.1 Bioinformatics databases

Multiple public databases serving different purposes are used in this work. They fall under the following four categories:

1. Sequence databases, which store the primary sequence of proteins.
2. Protein family databases, which curate classification systems for proteins and enzymes, usually based on evolution, homology, and conserved domains.
3. Protein annotation databases, which curate functionally and structurally important sequence signatures contained in proteins.
4. Tertiary structure databases, which store the resolved three-dimensional structures of proteins.

Sequence databases

UniProt is a central database that acts as a hub for most protein-related research on public data [1]. In just two years, over 65 million sequences were added to UniProt, representing a 50% increase. Also, UniProt stores entries for protein sequences deposited

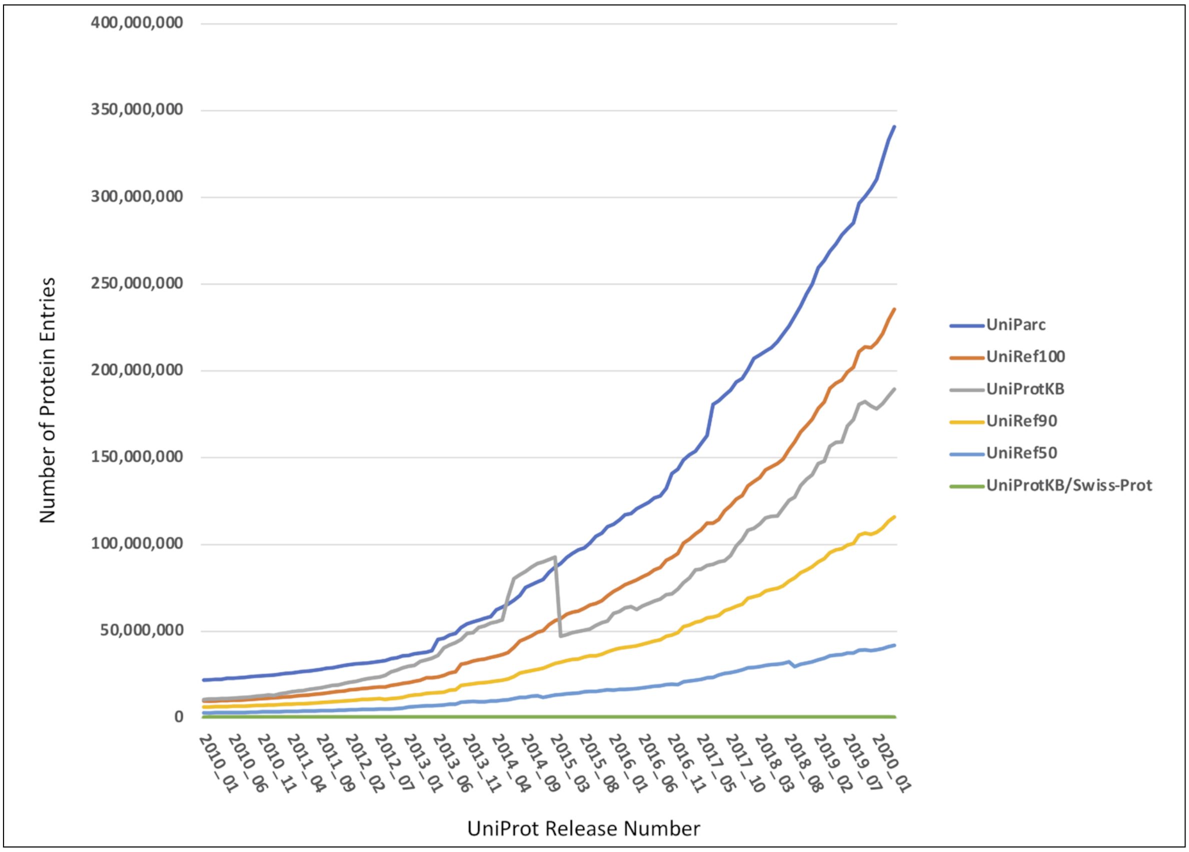


Figure 2.2: Plot showing the growth in the number of entries on UniProt in the last ten years, published by the UniProt consortium [1]. The growth of the manually curated branch of UniProt, Swiss-Prot, has massively lagged behind compared to the rest of the database.

by users along with a rich assortment of annotations when available, including functional annotation, subcellular location, post-translation modifications, protein-protein interactions, and so on.

UniProt is divided into two main resources: Swiss-Prot and TrEMBL. Swiss-Prot contains entries that are manually curated by experts, which helps provide a higher level of confidence about information contained in a protein entry. While TrEMBL does also get annotated, it is done using bespoke automatic annotation systems [1]. As would be expected given the characterisation bottleneck discussed in section 2.2.2, TrEMBL is, as of the June 2021 release, more than 380 times larger than Swiss-Prot, with 219,174,961 entries for the former, and 565,254 entries for the latter (Figure 2.2).

Protein family databases

Next, the Pfam database [33] was used to help retrieve enzymes based on shared common characteristics i.e. enzyme families. Specifically, Pfam families are built on profile-HMMs, which are probability-based models of alignments of related sequences. Such models can represent important features the sequences of an enzyme family has, such as conserved domains and functionally important residues. These HMMs can then be used to help detect sequences similar enough to be considered members of the same protein family. Each Pfam family is given an identifier e.g. PF00202 stands for the aminotransferase class-III family of enzymes. Much like UniProt, Pfam has had remarkable growth, increasing in size from 6109 families in 2004 to 18,259 in 2021 [51].

Protein annotation databases

The InterPro database [52] integrates sequence signatures from 13 other databases. The nature of the curated signatures varies, from smaller patterns of important residues like catalytic sites - from the Conserved Domain Database (CDD) [53] - to more generally conserved sequence-wide patterns - from Pfam [33]. These sequence signatures are detectable using sequence patterns and profile-HMMs, similarly to Pfam, with InterPro providing an automatic annotation tool called InterProScan [54].

Another primary annotation type is the enzyme commission (EC) system provided by the ENZYME database [55]. EC classes are a curated hierarchical classification system where numeric labels representing enzyme-catalysed reactions are assigned at four progressively more specific levels of functional detail, down to the level of substrate and reaction specificity. For example, the hierarchical breakdown of the EC class 2.6.1.1 is 2.-.-.-, which stands for transferases, 2.6.-.-, which stands for transferases that tranfer nitrogenous groups, 2.6.1.-, which stands for transaminases, and finally 2.6.1.1, which stands for aspartate transaminase. A drawback of EC numbers is that the substrate specificity of an enzyme might be known, but it may not have a complete EC number, as an EC number for that substrate has not been curated yet. Also, some complete EC numbers are still hierarchical in nature and correspond to entire classes of enzymes.

Tertiary structure databases

Finally, some of the more in-depth analyses of enzymes performed in this work used tertiary structures of enzymes as inputs. The primary bioinformatics resource for retrieving three-dimensional protein structures is the Protein Data Bank (PDB) [56]. Just like most bioinformatics databases, it has experienced high rates of growth over the years, going from 16,402 total structures with 2814 new releases in 2001 to 172,952 total structures with 14,029 new releases in 2020.

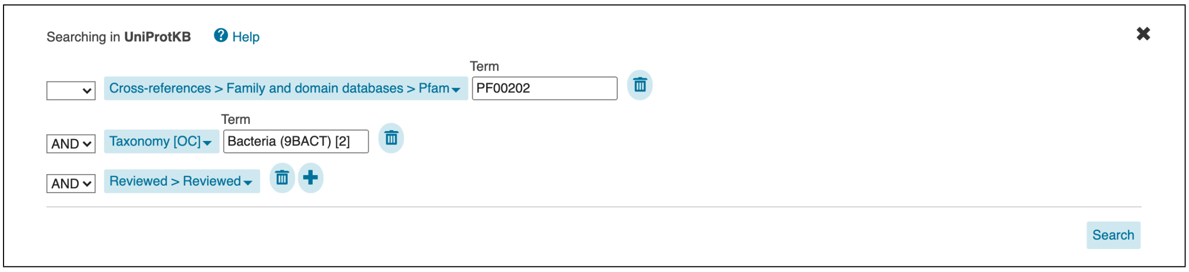
UniProt as a central hub for enzyme research

As was previously mentioned, UniProt acts as a central hub for most enzyme-related data retrieval from public databases. Indeed, even though all four databases described here are self-sufficient, UniProt integrates its protein entries with information from the other databases mentioned. For example, if an entry has a known tertiary structure on PDB, a cross-reference to its PDB entry is established. Similarly, UniProt entries contain any known Pfam family memberships and InterPro signatures. UniProt offers multiple avenues to querying and retrieving entries based on some search criteria, including simple but powerful search functionality on which conditional queries can be formed. For example, in Figure 2.3 a UniProt search query with three filters can be seen that retrieves entries that contain the Pfam ID PF00202 i.e. transaminases class-III, are bacterial in taxononmy, and that are reviewed i.e. from Swiss-Prot. UniProt also allows for programmatic access in the form of a application programming interface (API) and a SPARQL Protocol and RDF Query Language (SPARQL) endpoint [57]. All three of these interfaces are used extensively in this thesis to build datasets of enzymes that are annotated.

2.3.2 Enzyme families

An enzyme family has the following definition:

> “
>
> Functionally (or mechanistically) diverse superfamilies are evolutionarily related sets of enzymes that may be quite diverse in sequence, structure,



Searching in **UniProtKB** Help

Cross-references > Family and domain databases > Pfam — Term: PF00202

AND Taxonomy [OC] — Term: Bacteria (9BACT) [2]

AND Reviewed > Reviewed

Search

Figure 2.3: An example search query on UniProt using its advanced filtering options. This query in particular will retrieve entries of the transaminases class-III Pfam family (PF00202) that originate from bacteria on Swiss-Prot.

> and overall reaction, but share a conserved constellation of active site residues used for a common partial reaction or chemical capability [47]
>
> ”

As enzyme family evolution is still not well understood [47], for the sake of simplicity and consistency, enzymes in public databases were defined to belong to a certain family or superfamily if they were annotated with their respective Pfam ID i.e. if they are similar enough to the respective curated profile-HMM.

Specific enzyme families were chosen for diverse reasons which are explained in-depth in the chapters they are used. However, the general reasoning for choosing a family is the same: they are all very functionally diverse in terms of the number of different enzymatic reactions they catalyse, they are all populous and well-annotated enough on public databases that datasets of notable sizes can be built from them, and they all have applications as biocatalysts. Seven different Pfam family entries were used for dataset building in this thesis, which are:

1. Transaminase class I&II (PF00202), used in Chapter 3 [58].

2. Short-chain dehydrogenase (PF00106), used in Chapter 3 [59] .

3. Enoyl-CoA hydratase/isomerase (PF00378), used in Chapter 3 [60].

4. Transaminase class III (PF00155), used in Chapters 3 and 4 [58] .

5. Radical SAM (PF04055), used in Chapter 4 [61] .

6. Aldehyde dehydrogenase (PF00171), used in Chapter 4 [62].

7. Aldo/keto reductase (PF00248), used in Chapter 5 [63].

2.3.3 Bioinformatics tools

Many state-of-the-art bioinformatics tools exist, some of which can be split into four main categories:

1. Sequence alignment, which is the matching and comparison of one or more sequences to judge similarity and conservation of residues.

2. Sequence mining, which is the searching of sequences in a database similar to some given query sequence.

3. Sequence dataset visualisation, which is the displaying of sequences and the different relations between them, often with added annotation.

4. Enzyme analysis, which is the varied tooling used to assess and make predictions about individual enzymes, their structure, their function, and so on.

Sequence alignment

One major sequence alignment tool is the Basic Local Alignment Search Tool, or BLAST [40]. BLAST has been a seminal bioinformatics algorithm for over three decades, mainly as an efficient search tool that allows to find local sequence matches from a query sequence to a larger database. In this thesis, BLAST is mainly used not as a search tool but as a fast alignment tool to verify the pairwise similarity of one sequence to thousands of others. Specifically, the *blastp* command-line tool provided by the NCBI BLAST+ toolkit is used [64].

Another alignment tool is Needleman-Wunsch [65], which is a dynamic programming algorithm for pairwise alignments. Unlike BLAST, which is a local matching algorithm, Needleman-Wunsch is a global alignment tool, which finds the optimal alignment between two whole sequences rather than fragments. While BLAST is faster due to its nature as a heuristic, Needleman-Wunsch is more useful for judging the similarity of two sequences with full context. Needleman-Wunsch is mainly used for producing pairwise identity measures in this thesis, which are then used as inputs in various ways. Specifically, the *needle* command-line tool from the EMBOSS suite is used to perform these alignments [66].

Finally, Clustal Omega is another alignment tool that generates multiple sequence alignments (MSAs) [67]. MSAs are a standard data type in bioinformatics that help discern levels of similarity and patterns of conservation across large numbers of related sequences, unlike pairwise alignments which only compare two sequences. MSAs are used often to compare sequences in an enzymatic context, such as comparing active and binding site patterns. Clustal Omega is the latest iteration of the long-running Clustal MSA-generating tools [68], and is known to be fast at producing accurate alignments for even large sets of sequences. Specifically, the *clustalo* command-line implementation of ClustalOmega was used [69].

Sequence mining

One main sequence mining tool was used in this work: the HMM-based *hmmsearch*. *hmmsearch* is one of the many command-line tools contained in the HMMER suite [70]. HMMER is a bioinformatics toolkit that allows for the building and usage of profile-HMMs for various purposes like searching, which *hmmsearch* achieves. As required input, *hmmsearch* simply takes a profile-HMM and a file containing one or more amino acid sequences. Then, *hmmsearch* will find sequences in the input file that are similar enough to the given profile-HMM. Specifically, a sequence is considered similar enough if its resulting expect value - or e-value - is below a default or user-specified threshold. The e-value is the number of hits one would expect by chance to find in a database of a given size, with the lower the e-value the more significant the hit is likely to be. *hmmsearch* was used because the enzyme families of interest are summarised into profile-HMMs, as was described in section 2.3.2, which makes searching for hits of specific families more convenient [71].

Sequence dataset visualisation

The visualisation of entire enzyme family datasets in various ways is an crucial aspect of this thesis as it allows the deduction of important patterns in the relations between enzymes of a family. Two different state-of-the-art visualisation techniques were used often in this work, with the first one being phylogenetic trees [72]. Phylogenetic models allow for the visualisation and analysis of the evolutionary history of a set of sequences, and are powerful tools in attempts at understanding how diverse enzyme families evolved [73].

The second visualisation technique employed in this thesis is sequence similarity networks (SSNs). SSNs are homogeneous networks made up of sequence nodes - in the case of this thesis enzymes - and edges are made between such nodes if their sequence similarity is above some user-specified threshold [74]. While phylogenetic trees are very powerful at displaying evolutionary relationships between sequences, they are computationally heavy to produce for larger datasets. Also, SSNs are capable of displaying all the pairwise relationships of the sequences of a dataset, while trees cannot. However, they are not a direct substitute when trying to infer evolutionary history, as they are not built with such data in mind [74].

Enzyme analysis

During this thesis, individual enzymes often needed to be analysed in further contexts than sequence identity, phylogeny, and publicly available annotations. Two such contexts are employed here: tertiary structure analysis and functional annotation predictions.

The tertiary structure of a protein plays a key role in the type of function it fulfils, a well-known fact backed by multitudes of studies in the biological sciences [75]. While enzyme function is harder to deduce from sequence than for other proteins [44], the different tertiary structure 'folds' used by some enzyme family correlates well with their various functions [76]. Such a relationship between structure and function is often true even though both the tertiary structure and function of two enzymes can be still be highly similar at sequence identity levels as low as 16% [77] (Figure 2.4).

Therefore, when resolved tertiary structures were available for some enzymes - usually from the PDB database described in section 2.3.1 - these were used in analysis. However, when they were not available, a tertiary structure modelling tool called SWISS-MODEL [78] was used. SWISS-MODEL is a web-server that uses homology

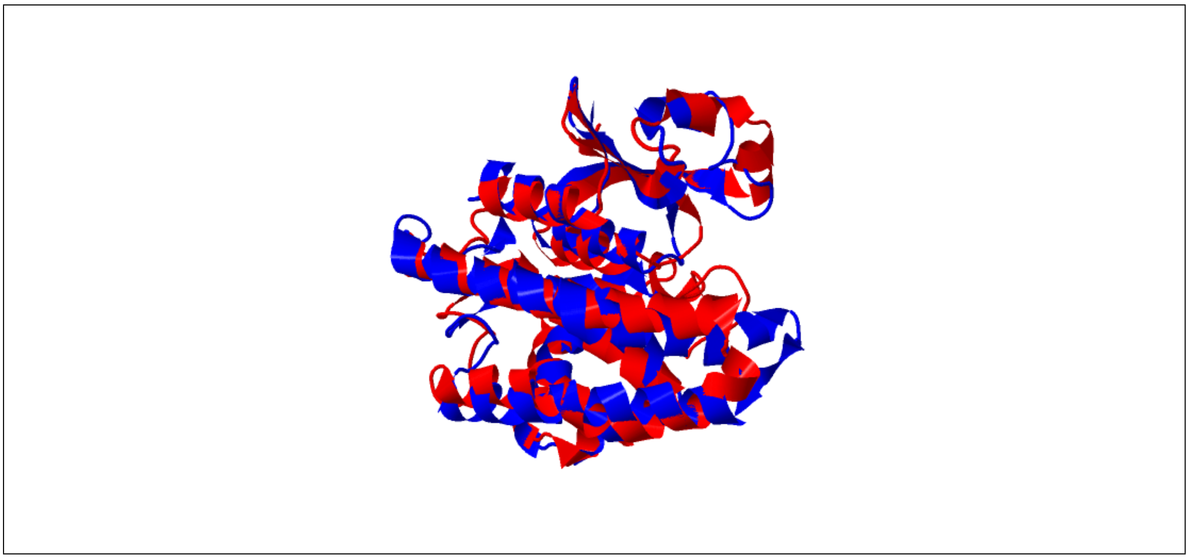


Figure 2.4: Tertiary structure overlap of two glutathione S-transferase homologues (EC class 2.5.1.18); 1GNW, which originates from *Arabidopsis thaliana*, and 1PGT, which is a human transferase. These two enzymes have high tertiary structure similarity, with a TM-score of 0.77, yet only share 16% sequence identity. This is just one example of the more complex sequence-structure-function relationship that enzymes have compared to other protein types.

modelling - which is the process of identifying structural templates to help build a model - to construct an accurate tertiary structure using a given query sequence as input. Homology modelling has proven to be a successful method of predicting accurate three-dimenstional models of proteins to help fill in the gaps induced by the bottleneck of manually resolving quality tertiary structures [79].

With a source of publicly available structures from PDB and the addition of accurate structure predictions using SWISS-MODEL, comparisons of tertiary structures were performed in this thesis. Specifically, much like the pairwise alignment of primary protein sequence is possible, as is the alignment of their three-dimensional structures using a tool called TM-align [80]. TM-align is an efficient and accurate tool for the alignment of protein structures, and produces an easy to interpret score called the TM-score . A TM-score between 0 and 0.3 represents a random structural similarity, while a TM-score between 0.5 and 1 was shown to be a good indicator for two proteins being in 'about the same fold' [81]. TM-align is used in this thesis often, in particular to investigate structural relationships of enzymes at low sequence identity.

Finally, the prediction of functional annotations of enzymes was performed using three

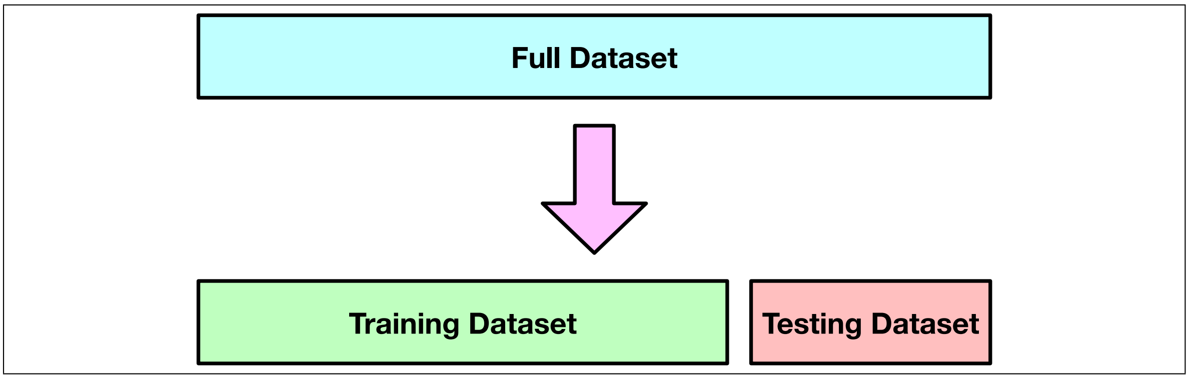


Figure 2.5: Diagram showing the common dataset split performed for machine learning exercises. Dataset of samples should be uniformly split into two both training and testing datasets, with the former being used for the learning process, and the latter for quality assessment purposes.

different tools, in particular when certain useful annotations were missing. First, InterProScan was used to detect InterPro signatures contained in unannotated enzymes, as was discussed in section 2.3.1 [54]. Second, DEEPre, which is a web-interface that allows for the machine-learning based prediction of EC classes using primary sequence as input, was used to provide more functional context, as DEEPre is highly accurate at least on the first three numbers of an EC class [28]. Finally, the iterative threading assembly refinement (I-TASSER) metaserver was used to provide in-depth analysis about structure and function with protein sequence as input [82]. I-TASSER is one of the state-of-the-art method for the prediction of protein structure, but also the identification of active and binding sites, and the likely catalytic functions an enzyme can catalyse.

2.4 Machine learning and heuristic optimisation

In this thesis, two major fields of computational research were used in substantial ways in novel applications for the creation of diverse enzyme panels: machine learning and heuristic optimisation. One definition of machine learning is the following:

> “A machine learning algorithm is a computational process that uses input data to achieve a desired task without being literally programmed to produce a particular outcome [83].”

Machine learning therefore tries to semi-automatically learn from some given input data how to solve a well-defined problem, through a process called 'training'. This field is often applied in situations where data have inherent functions that are complicated to discern by eye while still being theoretically mathematically learnable. The major types of problems solved by machine learning include classification problems, pattern recognition, artificial intelligence, among many others.

Importantly, one aspect of machine learning that is consistent across all its methods is the requirement for input data to be split in such a way that the performance of a trained model can be assessed on 'unseen' data. This split can be seen in Figure 2.5, and consists of separating a full dataset into a training dataset - which is used for the model to learn - and a testing dataset - which is used to assess the model's performance. There should be no overlap in samples between the training and testing dataset, and they need to be sampled in such a way that the proportion of labels are balanced similarly in both datasets. Also, every sample of such datasets should contain one or more labels that the machine learning models are tasked to predict.

One definition of heuristic optimisation is the following:

> “Heuristics are simple procedures, often guided by common sense, that are meant to provide good but not necessarily optimal solutions to difficult problems, easily and quickly [84].”

Optimisation problems are usually solved through the maximisation or minimisation of some objective function. Heuristic optimisation is therefore the process of making an educated guess at a solution that is of 'good' quality without necessarily being the best, with the main benefit of being computationally faster. Specifically, in an iterative process until some exit criterion is reached, neighbourhoods of solutions similar to the current solution state are generated, and decisions are made on which solution to 'move' to next. Heuristic algorithms are often applied to the non-deterministic polynomial-time hardness (NP-hard) class of problem, which includes most known genome assembly models [85].

Both of these fields are rich in approach and applications in bioinformatics. In this thesis, one method of each was used. Specifically, neural networks were utilised for a machine learning application, and the tabu search class of heuristic optimisation algorithms was used. A background level of knowledge on both of these paradigms is introduced in this section.

2.4.1 Neural networks

Neural networks are a set of machine learning models inspired by biological neural networks that are made up of artificial neurons and connections between them. Neural networks have been used in various state-of-the-art bioinformatics tools, from the secondary structure prediction tool PSIPRED [86], to the recently-released tertiary structure software AlphaFold [87], to the enzyme function prediction tool described in section 2.3.3, DEEPre [88].

While ongoing research into neural networks has generated diverse types for multitudes of applications, all share a core architecture, set of parameters, and learning logic. To give a background on this shared structure, the classic artificial neural network (ANN) is used as an example (Figure 2.6) [89].

In feedforward ANNs, there are three different types of neuron layers which are connected in a feedfoward manner by weighted edges. Each neuron has some value, usually between 0 and 1.

1. The input layer, which is the initial layer that data is passed to. The input data needs to be formulated in such a way that can be passed into this vector of input neurons.

2. The hidden layer - of which there can be more than one - the neurons of which mathematically transform input data using activation functions [90, 91] and the weights of connected edges.

3. The output layer, containing one or more neurons that should also be formulated in such a way that its neurons can be interpreted as a prediction of labels made by the neural network.

The weights of an ANN are initialised, usually to some random low values. Then, during training, input from the training dataset is passed to the input layer, one

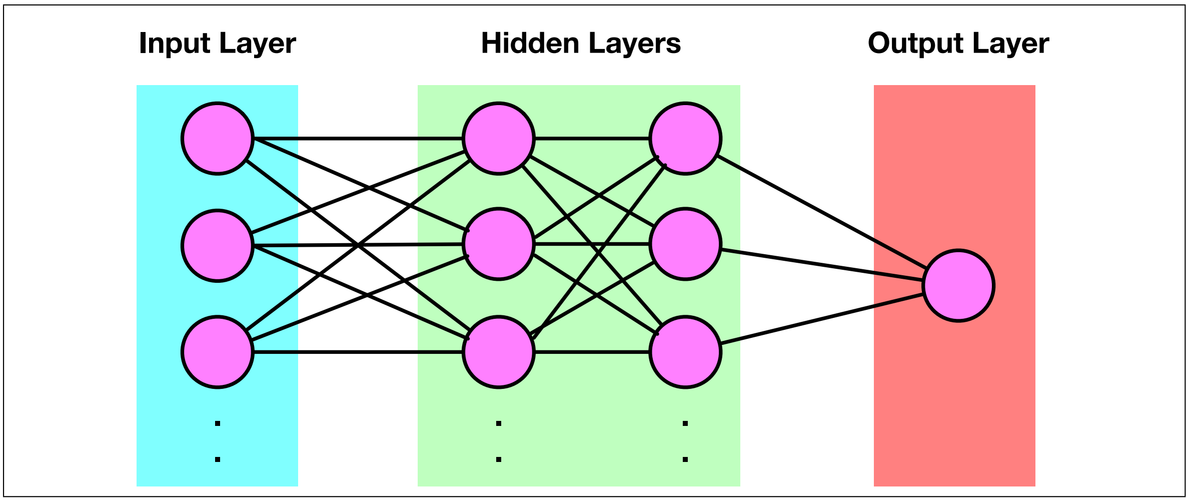


Figure 2.6: A diagram of the classic feedforward artificial neural network (ANN) model. It is made up of three main layers of neurons: an input layer, a hidden layer, and an output layer. During training, the error of predictions is backpropagated to the beginning of the ANN, changing the weights to make better predictions over the training process.

sample at a time. The values of the input layers are transformed by the hidden layers, before some initial prediction is made at the output layer. Based on the way the data was formulated, the predicted label is then compared to the true label using a loss function to calculate the error. There are many different existing loss functions like cross-entropy for classification problems [92] and the Huber loss for regression problems [93].

Then, using another class of function called an optimiser [94, 95], this error is 'back-propagated' in the reverse reaction of the feedforward ANN, changing the weights in the process. This iterative change in the weights is how neural networks fundamentally 'learn' how to solve the tasks they are given.

Neural networks research has progressed substantially over the years, and multitudes more types of neural networks than ANNs now exist. Convolutional neural networks for example are based on "the natural visual perception mechanism of the living creatures", and have proven successful at detecting important patterns and features contained in input data [96]. More recently, generative adversarial networks (GANs) have gained popularity for their use in the synthesis of novel data [97]. In this thesis, a variant of an autoencoder neural network [98] - which are powerful at the denoising and the dimensionality reduction of data - was described and used in Chapter 5.

2.4.2 Tabu search

Heuristic algorithms are often designed for bespoke purposes, or can be applied as black box models to multitudes of problems. In the latter case, these algorithms are called metaheuristics, which includes the tabu search class of algorithms. Tabu search was originally developed as a way of escaping local optima - a common challenge in optimisation problems - which is when an algorithm cannot ’climb’ out of a locally optimal solution in its current neighbourhood, resulting in solutions closer to the global optimum being unreachable [99, 100].

Tabu search algorithms achieve this by establishing a concept called ’tabu tenure’, whereby recently performed moves during the neighbourhood creation process are considered illegal moves to perform again, or ’tabu’ moves. Such moves are considered tabu for a defined amount of iterations - at which point the tabu status expires - which is the tabu tenure parameter. However, a predetermined condition called the ’aspiration criterion’, if reached by some tabu move, is the exception to the rule which will allow a tabu move to be performed. The combination of tabu tenure and the aspiration criterion therefore provides tabu search with ’short term memory’, which helps guide solutions away from local optima [99]. Through this short term memory, tabu search-based algorithms have been proven to be highly performing at solving many optimisation problems, including the travelling salesman problem [101], graph colouring [102], and the maximum diversity problem [103], the latter of which is used in Chapter 4 in a novel bioinformatics application.

2.5 Synthetic biology for enzyme research

Synthetic biology is the field of research that combines engineering concepts to the creation of useful biological systems [104]. In application, one of the principle aspects of synthetic biology is the modularisation of biology into usable ’parts’, like ribosome binding sites, promoters, and proteins, among others. The wet-lab expression and characterisation of enzymes requires a base level knowledge of synthetic biology and its approaches, which exist both *in silico* and *in vitro*. This section delves into the synthetic biology concepts utilised in this thesis and how they relate to enzyme

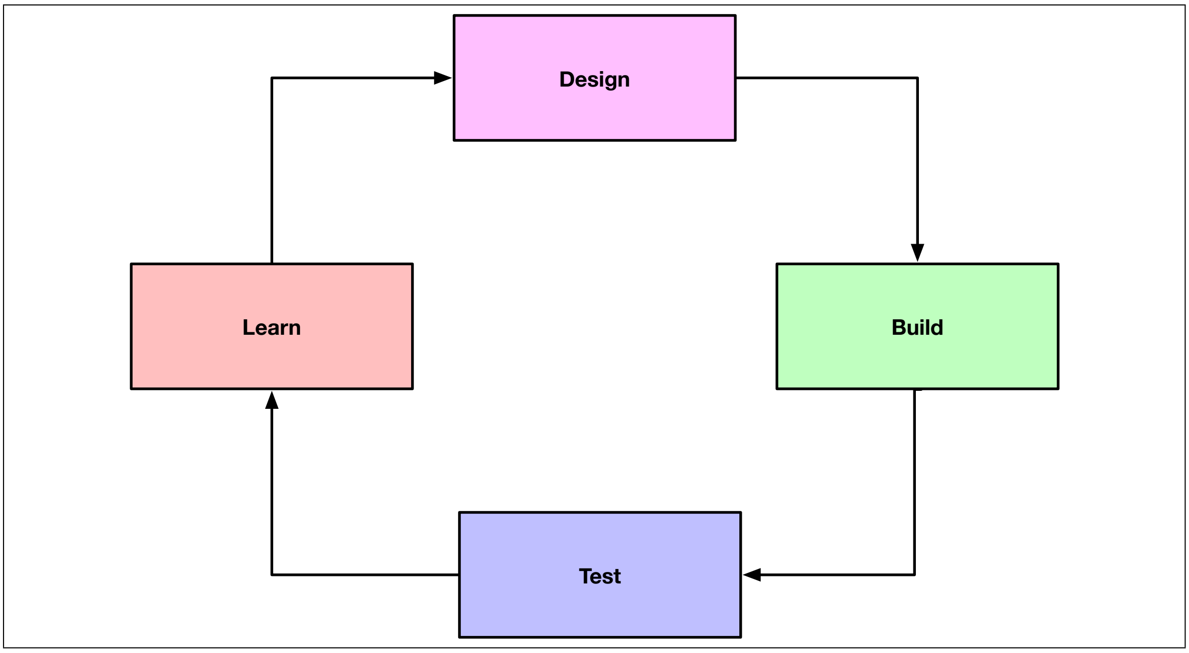


Figure 2.7: Diagram representing the engineering lifecycle, which is made up of four main stages: design, build, test, and learn. In synthetic biology, engineering concepts like the DBTL cycle are commonly used, in which it is referred to as the synthetic biology lifecycle. This lifecycle takes some synthetic biology project through the specification and design of some biological system, to its build and verification as a construct, to the testing of its performance, to the learning and changing of the original design based on the data produced in the test stage.

research.

2.5.1 The synthetic biology lifecycle

In many fields of engineering, a common pipeline for project management is the design-build-test-learn (DBTL) lifecycle [105]. This cyclic workflow has four main stages (Figure 2.7):

1. The design stage, during which an initial system is thought up to solve some problem and designed, often using computer-aided design tools.
2. The build stage, during which a design is implemented, with verification and quality assurance steps in place.
3. The test stage, during which a built design is tested for its original purpose, and assessed as to how well it achieves said purpose.

4. The learn stage, during which data about the performance of a tested design is used to gain insights on what can be improved its next iteration.

The DBTL lifecycle can also be applied to synthetic biology - in which it is referred to as the synthetic biology lifecycle - including to enzyme characterisation projects. For example:

1. At the synthetic biology design stage, an enzyme of interest could be reverse-translated into DNA, and then codon-optimised [106]. This DNA could then be added *in silico* to a plasmid design as a coding sequence.
2. At the build stage, synthesised gene fragments of the coding sequence of the enzyme could be cloned into plasmids in the laboratory. These plasmids would then be verified through sequencing before being transformed into a host organism as constructs.
3. At the test stage, the verified and built constructs are then used in enzyme characterisation experiments. Overexpression of the enzyme could be induced depending on the plasmid used [107], followed by an extraction of the protein specimen for assaying.
4. At the learn stage, the results of the test stage could be used to help optimise experimental conditions or even engineer the enzyme sequences themselves to be closer to optimal.

Using the DBTL lifecycle in synthetic biology is currently an important avenue of research for the field, with many novel methods having been developed to make it a more accessible workflow [108, 109]. The synthetic biology lifecycle played a key role in guiding the development of some of the approaches discussed in this thesis, as it provides an effective framework for an iterative characterisation of enzyme panels.

2.5.2 The Synthetic Biology Open Language

While the application of engineering concepts to synthetic biology has led to the modularisation of biological parts, this modularisation has become increasingly standardised.

For example, the international Genetically Engineered Machine (iGEM) competition [110] - which has taken place yearly since 2004 - requires the depositing of parts in the BioBrick standard [111] to the iGEM registry. In just ten years, over 12,000 parts resulting from the competition were added. The standardisation of synthetic biology concepts has continued to evolve, especially since the development of the Synthetic Biology Open Language (SBOL) [112].

SBOL is a standard that was originally introduced for the easy dissemination of DNA components, similar to other sequence formats like FASTA [113] and GenBank [114]. Since then, the capabilities of SBOL have expanded to include concepts more specific to synthetic biology, like the interactions between different components and combinatorial derivations. Changes to the SBOL specification are community-driven, through the creation of SBOL enhancement protocols (SEPs), which are only applied to the model if the SBOL community votes to do so as a majority.

While the SBOL specification is complex, it has been simplified with the recent release of SBOL3 [115]. However, the work performed in this thesis was done when SBOL2.3 was current, and is therefore the one described here [116]. Three key terms of the SBOL2.3 model need to be described:

1. The *TopLevel* class, which is an abstract superclass from which other classes inherit important properties from, like *name*, *description*, and *version*.
2. The *Collection* class, which inherits from *TopLevel*, is a class that allows for the grouping of other relevant *TopLevel* objects.
3. The *ComponentDefinition* class, which inherits from *TopLevel*, is a class that represents most biological entities, from DNA to RNA to proteins.

More interestingly for this research, SBOL has recently become capable of representing the entire synthetic biology lifecycle after the passing of SEP19 [117] and SEP21 [118]. SEP19 provided support for representing the build stage through a novel class called *Implementation*, while SEP21 added the classes *Experiment* and *ExperimentalData*. Also, SEP19 identified best practices for providing provenance information in SBOL, which is metadata crucial to reproducibility. This guidance is to use the following terms from the provenance ontology Prov-o [119]:

- The *Agent* class, which in SBOL is used to represent the person(s) and/or tool(s) that generate some *TopLevel* object.
- The *Plan* class, which in SBOL is used to represent the steps undertaken by some lab activity, like a lab protocol.
- The predicate *wasGeneratedBy*, which in SBOL is used to connect *TopLevel* objects to the *Agents* and *Plans* that generated them.
- The predicate *wasDerivedFrom*, which in SBOL is used to connect *TopLevel* objects of one DBTL stage to the *TopLevel* object they were based on. For example, a built *Implementation* would be connected to its original design *ComponentDefinition* through a *wasDerivedFrom* predicate.

As the learn stage can be performed through a distinction of modified designs using the *version* property of *TopLevel* classes, the SBOL model therefore has all of the necessary functionality for being able to represent the synthetic biology lifecycle and the data and metadata it produces. As the iterative characterisation of enzymes should ideally be performed using the lifecycle, SBOL was therefore the data standard used for relevant portions of this thesis.

2.5.3 Tooling gaps in SBOL

While the SBOL standard does possess the capacity for representing the DBTL lifecycle, tooling that helps perform the cycle using SBOL in the background is a key requirement due to its complexity. There are multitudes of tools for the design stage, like SBOLDesigner [120], CELLO [121], iBioSim [122]. ShortBOL [123] can also be used to create SBOL in a more abstract language. There are also repositories for designs like SynBioHub [109], and popular sequence-editing applications like Benchling (`https://www.benchling.com/`) can also output designs in SBOL format.

However, while SynBioHub can accept *Implementations* and *Experiments* existing in uploaded SBOL files, there is no current way of creating novel data of these types except through using the different SBOL libraries like *sboljs* [124], which is not accessible, in particular to those without a programming background. Flapjack is an SBOL-powered interface that aims to facilitate the transition from the test and learn stages, but does not physically store constructs from the design and build stages [125].

There is therefore a major tooling gap in SBOL, especially in the later stages of the synthetic biology lifecycle like build and test. Even though the data model itself can technically represent the DBTL cycle, the methods to support the model do not yet exist. The implication from this lack of tooling that is relevant for this thesis is a reduced inefficiency in the iterative characterisation of enzyme panels. This significant limitation was therefore addressed in this thesis, specifically in Chapter 6.

2.6 Summary and conclusion

The use of enzymes as biocatalysts in various industries continues to grow in popularity. This growth is due to various benefits in the replacing of standard chemical synthesis methods with an enzyme, including higher chemical yield and regio/stereoselectivity, simplified pathways, and greener chemistry. However, while the amount of discoverable enzyme diversity has undoubtedly grown thanks to the advent of next-generation sequencing, this diversity is merely **discoverable**, not **discovered**. Indeed, most enzymes on public databases are only loosely annotated using automatic systems, with a large proportion not being annotated at all.

Therefore, initiatives that attempt to mass-characterise enzyme families in the lab are of particular value, as such assays are currently the only methods that can confidently reveal the function of enzymes. However, the current approaches to the selection of diverse enzyme panels to be taken into the lab is limited without a unified framework. The major limitations include an over-reliance on sequence identity for the assignment of homology, a high burden of knowledge for more in-depth analyses of diversity, analyses which are complex and time-consuming. Also, while the use of the synthetic biology lifecycle would be an ideal framework for the iterative characterisation of enzymes in the laboratory, there are significant tooling gaps to make its use accessible in reality.

All of these limitations have left research gaps that could be filled to help optimise the process of generating diverse panels of enzymes for characterisation in the laboratory. The tackling of these gaps has resulted in the identification of two principle aims for this thesis:

1. The development of new computational methods for building diverse sequence panels from enzyme families
2. The building of tools that promote an integrated and iterative framework for the characterisation of enzyme families in the laboratory

In this thesis, methods that achieve these aims are introduced and discussed, starting with a method of analysing enzyme families that is not based on raw sequence similarity in Chapter 3.

3

Functional analysis of enzyme families using coevolution similarity networks

Contents

3.1 Introduction

The functional analysis of protein families is essential for the application of enzymes in biotechnology, as was mentioned in section 2.2. Confidently assigning functional annotations to enzymes and their families can lead to a better understanding of enzyme evolution and enzyme diversity [126] underpinning many areas of functional genomics. This understanding can help better pinpoint the sequence space of an enzyme family that needs further study, leading to better results when building panels to be tested in the laboratory, as was discussed in the Background (section 2.2.2).

However, the functional analysis and annotation of enzymes and their families is difficult. Amino acid sequence similarity is the most commonly used evidence for annotating protein and enzyme function [2, 127], usually through the identification of homologous relationships, between which annotations are then transferred. However, as described in section 2.2.3, such sequence identity methods lose power when applied to the conservation of enzyme function, especially at the substrate specificity level.

The transfer of annotation relating to the substrate specificity of enzymes is also made particularly difficult by the overall low quality of existing annotation on public databases. As of the 13/02/2019 release of UniProt [128], TrEMBL contains over 140 million sequences, while the manually-curated Swiss-Prot only contains around 550,000, of which 238,254 are entries annotated with ”catalytic activity’ [129, 130]. Only 10,921 (4.5%) of the ”catalytic activity’ Swiss-Prot entries have their catalytic activity annotations supported by evidence that is ”manually curated information for which there is published experimental evidence”. The majority, 179,784 (74.5%) entries have had their catalytic activity automatically assigned by Uniprot ”sequence models”, manually-curated protein family profiles to which new entries are matched [131, 132].

While these profiles are manually curated and kept up to date, sequence similarity is ultimately the core metric behind the assignment of annotations. The proportion of evidence for annotation that is laboratory-based, and therefore which can be confidently trusted, is consequently low, which has led to the erroneous propagation of catalytic activity assignments, which is a known problem on public databases [133, 134]. It can also lead to missing annotations, as enzymes can be notably promiscuous in terms of their substrate activity.

The rapid increase in the availability of protein sequences has also lead to an increase in the size of enzyme protein families. It is now necessary to study the structure and functional diversity of large enzyme families at a much greater scale than previously required, including for the optimisation of enzyme panel selection for characterisation in the laboratory. Standard phylogenetic approaches to family assignment become computationally challenging when dealing with large protein families [135], to both produce but also interpret, leading to unoptimised selection approaches like those discussed in section 2.2.3.

The idea of building networks of proteins based on raw sequence similarity has gained popularity as a rapid method for gaining a visual and structural overview of large protein families. Approaches such as Sequence Similarity Networks (SSN) have been shown to be useful for providing overviews of the functional diversity of enzyme families and superfamilies [47, 74, 136]. These networks have been demonstrated to be valuable for visualizing functional trends across protein superfamilies, relying on the assumption that structural or sequence-based similarity implies functional similarity, while being computationally less heavy than phylogenetic trees. SSNs, just like the other methods discussed in section 2.2.2, can therefore be used for the analysis of sequence datasets in the preparation of diverse enzyme panels for the laboratory. However, SSNs also suffer from the drawbacks associated with sequence based annotation mentioned in section 2.2.3 of the Background.

Therefore, new approaches are required to more confidently assign functional homology of enzymes, annotate enzyme substrate specificity, and perform functional analyses of large and diverse enzyme protein families. Specifically, some method that can complement sequence-similarity based methods to fill in homology gaps at low sequence identity would be of high impact, including for the analysis of enzyme datasets for panel creation.

3.1.1 Residue-residue coevolution

The analysis of residue-residue coevolution is another useful method for searching for functional conservation in protein sequence and structure. Two amino acid residues

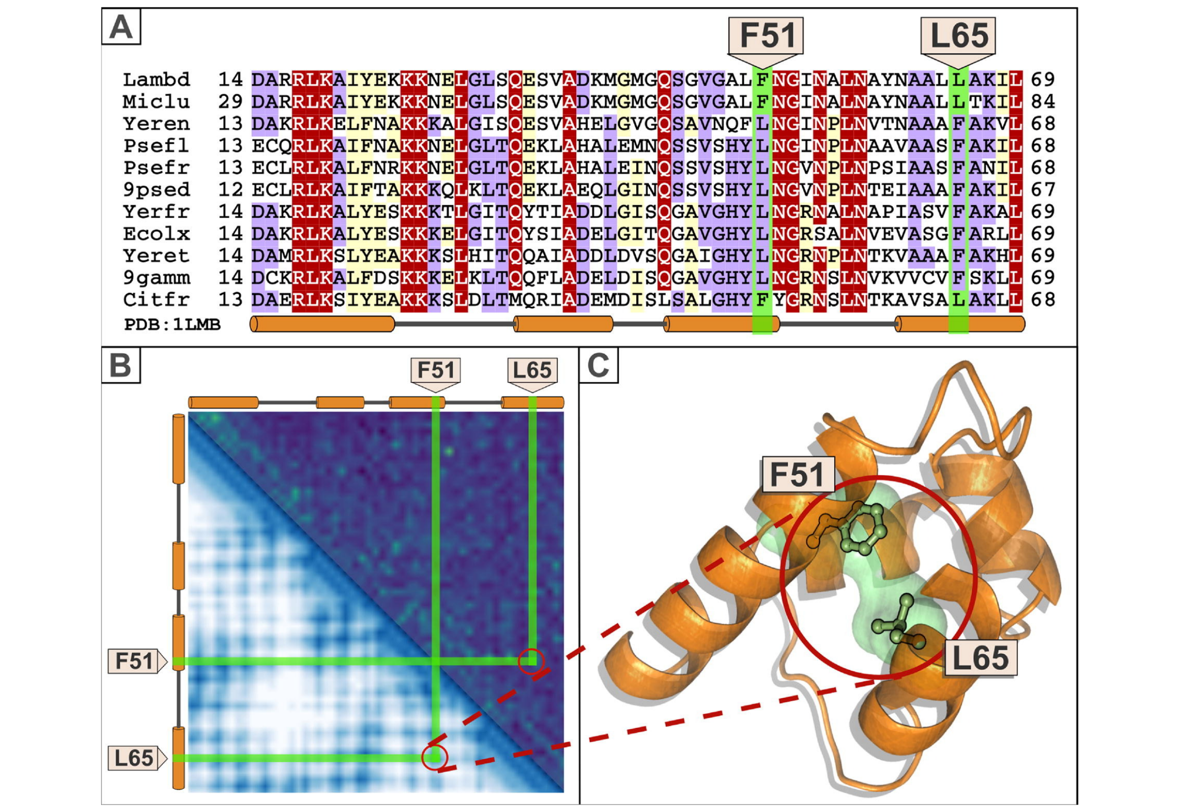

A
F51
L65
Lambd 14 DARRLKAIYEKKKNELGLSQESVADKMGMGQSGVGALFNGINALNAYNAALLAKIL 69
Miclu 29 DARRLKAIYEKKKNELGLSQESVADKMGMGQSGVGALFNGINALNAYNAALLTKIL 84
Yeren 13 DAKRLKELFNAKKKALGISQESVAHELGVGQSAVNQFLNGINPLNVTNAAAFAKVL 68
Psefl 13 ECQRLKAIFNAKKNELGLTQEKLAHALEMNQSSVSHYLNGINPLNAAVAASFAKIL 68
Psefr 13 ECLRLKALFNRKKNELGLTQEKLAHALEINQSSVSHYLNGVNPLNPSIAASFANIL 68
9psed 12 ECLRLKAIFTAKKKQLKLTQEKLAEQLGINQSSVSHYLNGVNPLNTEIAAAFAKIL 67
Yerfr 14 DAKRLKALYESKKKTLGITQYTIADDLGISQGAVGHYLNGRNALNAPIASVFAKAL 69
Ecolx 14 DAKRLKALYESKKKELGITQYSIADELGITQGAVGHYLNGRSALNVEVASGFARLL 69
Yeret 14 DAMRLKSLYEAKKKSLHITQQAIADDLDVSQGAIGHYLNGRNPLNTKVAAAFAKHL 69
9gamm 14 DCKRLKALFDSKKKELKLTQQFLADELDISQGAVGHYLNGRNSLNVKVVCVFSKLL 69
Citfr 13 DAERLKSIYEAKKKSLDLTMQRIADEMDISLSALGHYFYGRNSLNTKAVSALAKLL 68
PDB:1LMB
B
F51
L65
F51
L65
C
F51
L65

Figure 3.1: Residue-residue coevolution is a good proxy for residue-residue contact, published by Sanchez-Pulido and Ponting [2]. A) An MSA of homologous proteins from the lambda repressor family. The positions F51 and L65 are found to coevolve, as a mutation in one position leads to a mutation in the other. B) A heatmap showing the correlations between different residues. F51 and L65 are seen to correlate, and therefore likely coevolve. C) One of the secondary structures of a lambda repressor. The coevolving F51 and L65 residues are in close contact.

are said to coevolve if an influential substitution in one residue is counteracted by a substitution in the other residue [137]. This phenomenon occurs due to evolutionary pressure on a protein to retain structural stability. Indeed, residue-residue coevolution is known to correlate particularly well with residue-residue contact, and has successfully been used to discern the maps of contacts of proteins [137] (Figure 3.1). Residue-residue coevolution information has been used to infer spatial constraints for residues for the purposes of predicting tertiary structures [138, 139] in single proteins and protein-protein interactions between two or more proteins [140].

Patterns of coevolution have also been used in a functional context. Coevolution patterns are often represented as homogeneous residue-residue coevolution networks, in which proteins are represented as networks, where nodes represent residues and edges exist between nodes when the residues involved are likely to coevolve. Such networks have been successfully used for identifying functionally important residues [141–144]. With the emergence of more accurate methods for computing coevolution metrics on a large scale, largely based on direct coupling analysis (DCA), [3, 139, 145, 146], it is now possible to produce coevolution data for a set of enzymes and to build residue-residue coevolution networks for pairs of enzymes on a larger scale than has previously been possible.

Pairwise coevolution data have been used to analyse the functions of individual enzymes, but to the author's knowledge there has been little or no use made of protein coevolution data as a similarity metric between proteins in protein family-wide functional analyses. As coevolution patterns contain information important for the determination of protein function, it is expected that functionally similar enzymes, specifically with respect to substrate specificity, will share similar coevolution patterns. The hypothesis explored in this chapter is therefore the following:

> “An all-vs-all comparison of residue-residue coevolution patterns in an enzyme family can be used to perform a functional analysis of that family.”

In this chapter, a novel method is introduced for inferring functional relationships between enzymes called coevolution similarity networks (CSNs). CSNs have a graph definition similar to that of sequence similarity networks, where nodes represent proteins, and edges exist between nodes if their coevolution patterns are similar, with the similarity metric based on a user-defined threshold. CNSs also display a network-based overview of the structure and functional relationships within a protein family, and provide a complementary approach for the functional annotation of enzymes. Such an approach would be of high-value for the analysis of datasets of enzyme families, including for the purposes of delineating functional homologues in a way that is useful for enzyme panel creation.

This chapter explains how CSNs can be used to represent the distribution of functional diversity of a set of enzymes across a network in terms of substrate specificity, in a way that groups functionally similar nodes, in a similar but complementary fashion to SSNs. A comparison of the network structure and performance of CSNs and SSNs was performed for three different annotated enzyme datasets from Swiss-Prot: transaminases class III, transaminases classes I/II, and short-chain dehydrogenases. Specifically, the family structure was explored for each dataset, and the predictive power of the networks for the purposes of enzyme substrate specificity was tested using a label propagation experiment. Also, CSNs were also used to reveal discrepancies in functional annotation in a fourth dataset made up of enzymes from the crotonase family.

3.2 Materials and methods

3.2.1 Enzyme datasets

Four datasets from different enzyme families were used in this work. The first comprised 241 transaminase class III enzymes, the second was a set of 986 transaminase class I/II enzymes, the third was a set of 142 enzymes from the short-chain dehydrogenases/reductases (SDR) family, and the fourth dataset was made up of 99 crotonases. All four datasets were built using data from Swiss-Prot [128]. These datasets were chosen since they are well annotated, functionally diverse, and are relevant to fields such as the biocatalyst industry.

The datasets were built by searching Swiss-Prot for prokaryotic entries that contain the PFAM identifiers for their respective families: PF00202 for the transaminase class III enzymes; PF00155 for the transaminase class I/II enzymes; PF00106 for the SDRs; and PF00378 for the crotonases [33]. These datasets are referred to as Trans241, Trans986, SDR142, and Croto99, respectively, in the remainder of this paper. A table showing the distribution of all the EC classes across the four datasets can be seen in table 3.1.

Datasets from Swiss-Prot were used since the functional annotations of proteins in this database are more accurate than those from other sources [133]. An important annotation type assigned to most of these entries is the Enzyme Commission (EC) number [55]. EC numbering is a hierarchical classification system that assigns numeric labels to an enzymatic reaction at four progressively more specific levels of functional detail, down to a level of detail where the substrate specificity can be implicitly derived from the reaction name.

Table 3.1: Table showing the distribution of EC classes for the Trans241, Trans986, SDR142, and Croto99 datasets.

	EC	Count	EC	Count			EC	Count	EC	Count
Trans241	5.4.3.8	98	2.6.1.105	1		**Trans986**	2.6.1.9	464	2.3.-.-	2
	2.6.1.11	82	2.6.1.82	1			2.3.1.47	294	2.6.1.5	2
	2.6.1.62	22	2.6.1.111	1			2.6.1.83	95	2.6.1.66	2
	2.6.1.17	10	2.6.1.77	1			2.6.1.-	39	2.6.1.39	2
	2.6.1.19	7	5.1.1.15	1			2.6.1.1	31	2.6.1.17	2
	2.6.1.76	7	2.8.1.6	1			2.3.1.37	12	2.6.1.117	1
	2.6.1.22	6	2.6.1.95	1			2.6.1.79	9	2.6.1.103	1
	2.6.1.-	5	2.6.1.13	1			2.3.1.-	9	2.6.1.107	1
	2.6.1.81	4	5.1.1.21	1			2.6.1.2	8	2.6.1.88	1
	2.6.1.93	3	4.1.1.64	1			4.4.1.13	7	4.2.1.145	1
	2.6.1.18	2	2.6.1.113	1			2.6.1.57	5	2.6.1.14	1
	2.6.1.48	2	2.6.1.94	1			2.3.1.29	5	2.6.1.38	1
	2.6.1.36	2					4.1.1.81	5	2.6.1.84	1
							2.6.1.78	4		
SDR142	1.-.-.-	54	1.1.1.69	2						
	1.3.1.87	20	1.3.1.33	2						
	1.1.1.381	6	1.1.1.50	1						
	1.1.1.298	6	1.1.1.395	1						
	1.3.1.56	6	1.1.1.n4	1						
	1.1.1.-	5	1.1.1.325	1						
	1.3.1.-	5	1.3.1.49	1		**Croto99**	4.2.1.17	27	4.1.1.-	1
	1.1.1.36	4	5.1.3.34	1			4.2.1.149	25	4.2.1.155	1
	1.1.1.333	4	1.1.1.256	1			4.1.3.36	19	4.2.1.100	1
	-.-.-.-	4	1.1.1.201	1			-.-.-.-	4	4.1.2.41	1
	1.1.1.304	3	1.3.1.28	1			3.8.1.7	3	4.2.1.101	1
	1.1.1.340	3	1.1.1.52	1			4.2.1.150	3	2.3.1.226	1
	1.1.1.313	3	1.2.1.n2	1			3.7.1.21	3	4.2.1.-	1
	1.1.1.320	2	1.3.1.19	1			5.3.3.14	2	4.-.-.-	1
	1.1.1.276	2	1.1.1.56	1			1.13.11.80	2	4.1.2.44	1
	1.1.1.140	2	1.1.1.30	1			3.7.1.18	1	1.1.1.35	1
	1.3.1.25	2					5.3.3.18	1		

3.2.2 SSN construction

Other researchers have used BLAST-based approaches to search for sequences and build datasets, from which SSNs are then built by thresholding the e-value [74]. However, the e-value produced by a BLAST search depends on the size of the dataset, and BLAST therefore cannot be applied consistently across the four datasets. Therefore, SSNs were generated using global pairwise alignments [65] in an all-vs-all fashion, producing a sequence identity matrix for all four datasets. This matrix was then used along with an identity threshold value to build.

3.2.3 Residue-residue coevolution network construction

The coevolution data for each enzyme were produced using CCMpred [3], following the recommended protocol [147]. The result was a residue-residue coevolution matrix for every enzyme across all four datasets (Figure 3.2). Then, for studies on the residue couplings that are most likely to coevolve, the developers of CCMpred suggest ranking all the couplings and picking the top N with the highest score. In this work, N was chosen to be an arbitrarily high number (600 for SDR142, 700 for Trans241, Trans986, and Croto99) in order to ensure the inclusion of coevolving pairs that are unique to functional subclasses of the family. From the selected pairs of residues, created residue-residue coevolution networks were created for each protein. In these networks nodes represent residues, and edges exist between the residues of a coevolving pair within a single protein.

3.2.4 Residue-residue coevolution network mapping

To compare the residue-residue coevolution networks of individual enzymes, a method for comparing coevolving pairs of enzymes and estimating whether these pairs are equivalent in terms of positioning in homologous proteins was developed. The amino acid sequences of proteins in each dataset were aligned using a multiple sequence alignment (MSA) algorithm, to establish the relative positions of coevolving residues. It was then assumed that if two residues of a coevolving pair of one enzyme aligned to two residues from another enzyme that also coevolve, they were equivalent coevolving

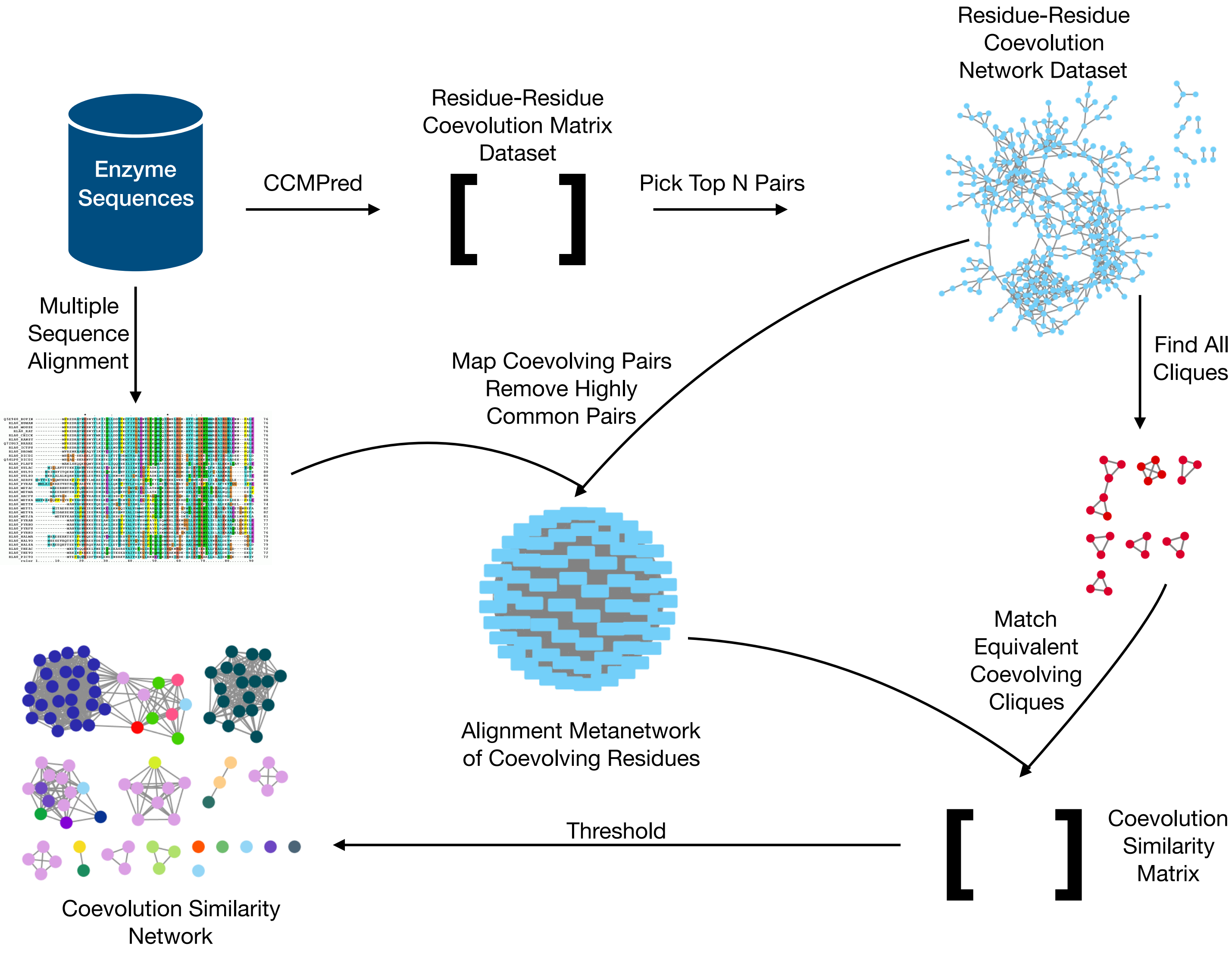

Enzyme Sequences
CCMPred
Residue-Residue Coevolution Matrix Dataset
Pick Top N Pairs
Residue-Residue Coevolution Network Dataset
Multiple Sequence Alignment
Map Coevolving Pairs Remove Highly Common Pairs
Find All Cliques
Match Equivalent Coevolving Cliques
Alignment Metanetwork of Coevolving Residues
Coevolution Similarity Matrix
Threshold
Coevolution Similarity Network

Figure 3.2: The workflow used to produce a Coevolution Similarity Network (CSN) for a dataset of protein sequences. From a set of amino acid sequences, the workflow first produces networks representing residue-residue coevolution within an individual sequence, using CCMPred [3]. A mapping of all of the equivalent coevolving pairs is performed for all sequences after filtering common pairs using a multiple sequence alignment, resulting in an alignment metanetwork of coevolving residues. Using this mapping, an all-vs-all comparison of the residue-residue coevolution networks is performed by matching each network's cliques, producing a coevolution similarity matrix. A threshold is then set for this matrix, and the final output is a CSN.

pairs (ECPs). Clustal-Omega [67] was used to align all of the sequences of a set into MSAs, one for each of the three datasets.

For each dataset an intermediate alignment meta-network was created, in which nodes represented the coevolving residues of all of the residue-residue coevolving networks of a dataset, and edges were made between nodes if they aligned according to the MSA; that is, if they were ECPs.

3.2.5 Clique-based residue-residue coevolution network comparison

For each residue-residue coevolution network, all the cliques were computed using NetworkX [148]. Cliques are a concept in network theory that describe groups of nodes in a network that are fully connected. Cliques are relevant in this work because ”in the context of coevolution, a clique represents a set of residues wherein each residue covaries with all of the others” according to Lee and coworkers [144].

Then, a square scoring matrix was produced by matching the cliques of the residue-residue coevolution networks. If a clique in enzyme A had X coevolving pairs, and all X coevolving pairs had ECPs in enzyme B they were considered to be equivalent coevolving cliques, and the score between A and B was incremented by 1 (Figure 3.3).

The clique similarity scores were then normalised to values between 0 and 1 by transforming them into Jaccard similarity scores, in which A and B are the number of cliques of the residue-residue coevolution networks for enzymes A and B (Equation 3.1). These scores, which are termed as ”coevolution similarity” or CS scores henceforth, were used to produce CSNs in which nodes are enzymes and edges exist between two nodes if their similarity value is above a user-defined threshold.

$$\mathrm{CS} = \frac{|A \cap B|}{|A| + |B| - |A| \cap |B|} \tag{3.1}$$

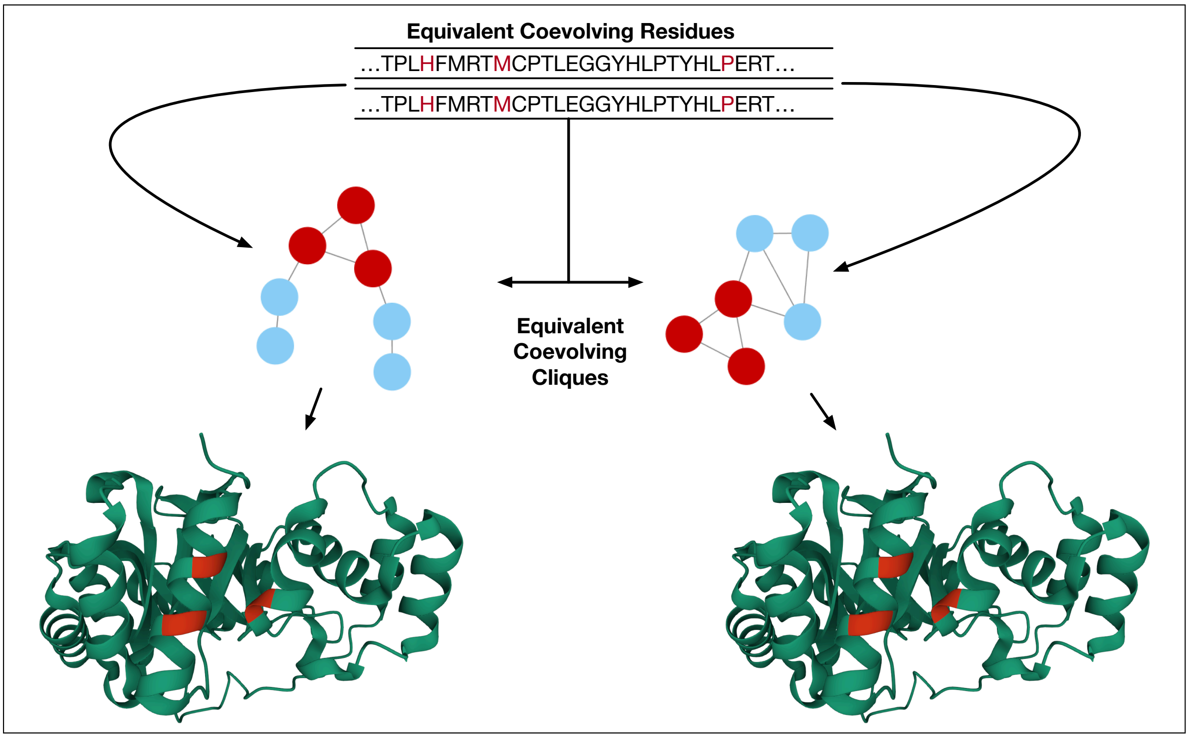

Equivalent Coevolving Residues
...TPLHFMRTMCPTLEGGYHLPTYHLPERT...
...TPLHFMRTMCPTLEGGYHLPTYHLPERT...
Equivalent Coevolving Cliques

Figure 3.3: The mapping of ECPs onto equivalent coevolving cliques. When all the residues of a clique of coevolving residues are are ECPs within another protein's cliques, then these are considered as equivalent coevolving cliques.

3.2.6 Filtering of family-wide Equivalent Coevolving Pairs

Some coevolving pairs are expected to be general to the family as a whole [149]. ECPs that were present in the majority of sequences in a dataset would therefore be uninformative for the purpose of discriminating function at the specificity level. This is especially true as while ECPs common to a whole family are likely to be important for structure and function, "other positions may be conserved only within particular subfamilies. These subfamily-specific residues are likely to define the specific functionality of that subfamily, such as forming three-dimensional clusters that make-up ligand- and/or protein-binding sites or allosteric chains of residues" [149].

If coevolution similarity scores function similarly to sequence similarity, it can be assumed that the higher the score between two enzymes the more likely they are to perform similar enzymatic functions. However, if the coevolution similarity score tends closer to 1 on average because a family shares a high amount of the total coevolving pairs across all its sequences, then it becomes harder to threshold and build CSNs in a way that is specific enough to substrate specificity.

For example, imagine a gold-standard similarity network existed for an enzyme dataset with the following properties and assumptions:

- Each enzyme has at most one EC number, with no promiscuous enzymes
- The enzymes themselves are all correctly annotated with EC numbers
- Only edges between enzymes within the same EC number are created, with no edges between nodes of differing EC numbers

The gold-standard similarity network for the Trans241 dataset would have 7545 edges spread across its 25 different EC numbers. When a CSN is created using the methods described previously in this section, without any filtering of the common ECPs, the threshold that comes closest to creating a CSN of this size is 0.856, with 7578 edges in total (Figure 3.4). Because the average coevolution similarity is already so high, the similarity threshold required to get to a CSN as specific as the gold standard is therefore also high.

This aspect of CSNs makes it difficult to pinpoint thresholds that make linkages specific to substrate specificity, as any specificity-determinant ECPs that are common only to certain EC classes are drowned out by the ECPs common to the family as a whole. Indeed, if the threshold for the Trans241 CSN is lowered by as little as 0.006 to 0.850, there is an increase in the number of edges of 13.5%, to 8682 edges in total for the CSN. Such a drastic change in the number of edges from such a small threshold difference makes it more difficult to produce a CSN that models the linkages in the dataset in a meaningful way.

This challenge can also be visualised with heatmaps (Figure 3.5). In the unfiltered heatmap, it can be seen that overall coevolution similarity is high, with an average score of 0.69. While there are clusters that are brighter than others, it is more difficult to parse because the background is already bright, which makes it more difficult to judge how high the similarity should be to associate two enzymes as similar functionally.

Therefore, coevolving pairs that occurred in a high proportion of the dataset were filtered out from the alignment network. This filtering of common coevolving pairs

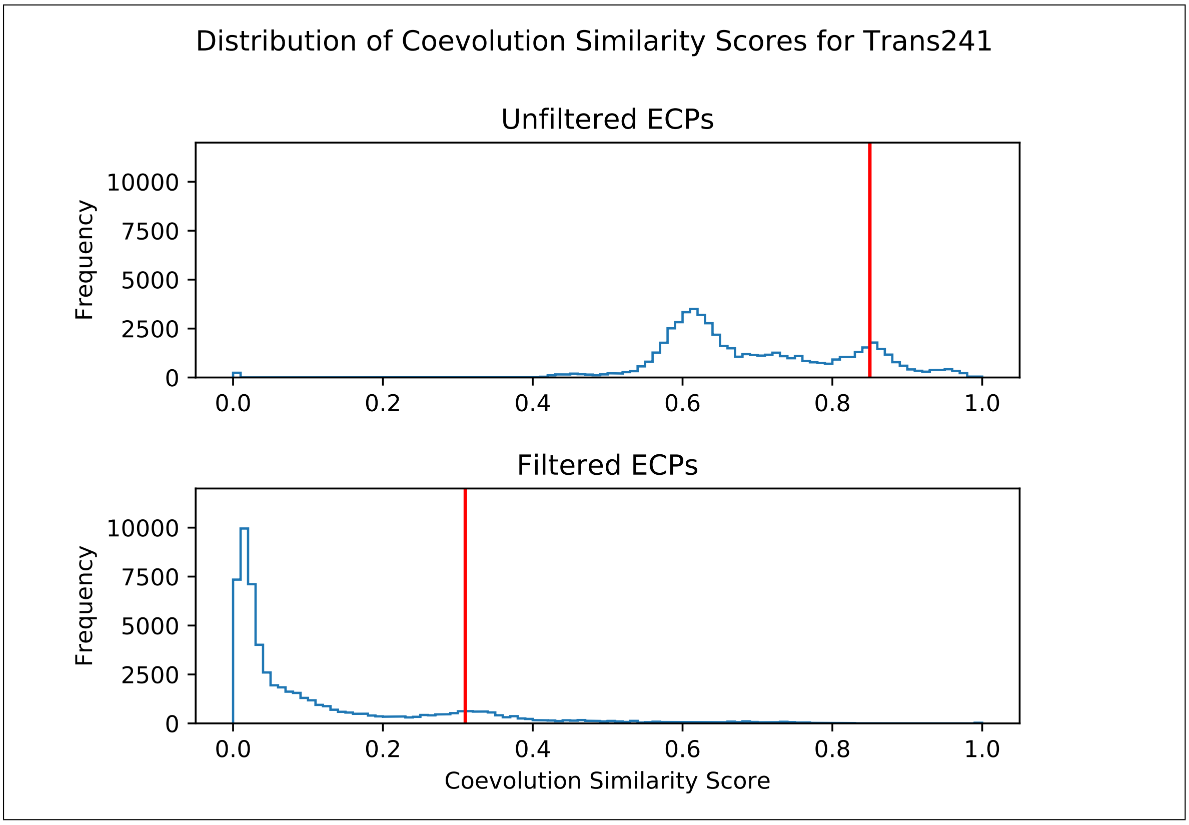


Figure 3.4: Distributions of coevolution similarity scores for the Trans241 dataset before and after filtering of common ECPs.The red line represents the number of edges the gold-standard similarity network would have. With the distribution being more skewed to the left after the removal of ECPs that are likely to be common to the family as a whole, the coevolution similarity score is more impactful when it is actually high, making it easier to discern more specific functional linkages.

has the effect of amplifying the contribution of the pairs more unique to functional subclasses during the comparisons of residue-residue coevolution networks. Such comparisons are more likely to reveal precise linkages at the substrate specificity level.

The result of filtering of family-wide coevolving pairs for the Trans241 dataset can be seen in Figure 3.4, where ECPs common to over 60% the dataset were discarded. The score distribution has been skewed to the left, with 79% of the coevolution similarity scores being below 0.2. The threshold necessary for reaching the gold standard similarity network size is 0.31, with 7580 edges, a far lower threshold than the unfiltered CSN at 0.85. With ECPs common to 60% of the dataset not being considered, it means that when two enzymes have a high coevolution similarity score it is likely more impactful, resulting in CSNs that can more easily distinguish coevolution linkages that represent functional diversity in the family.

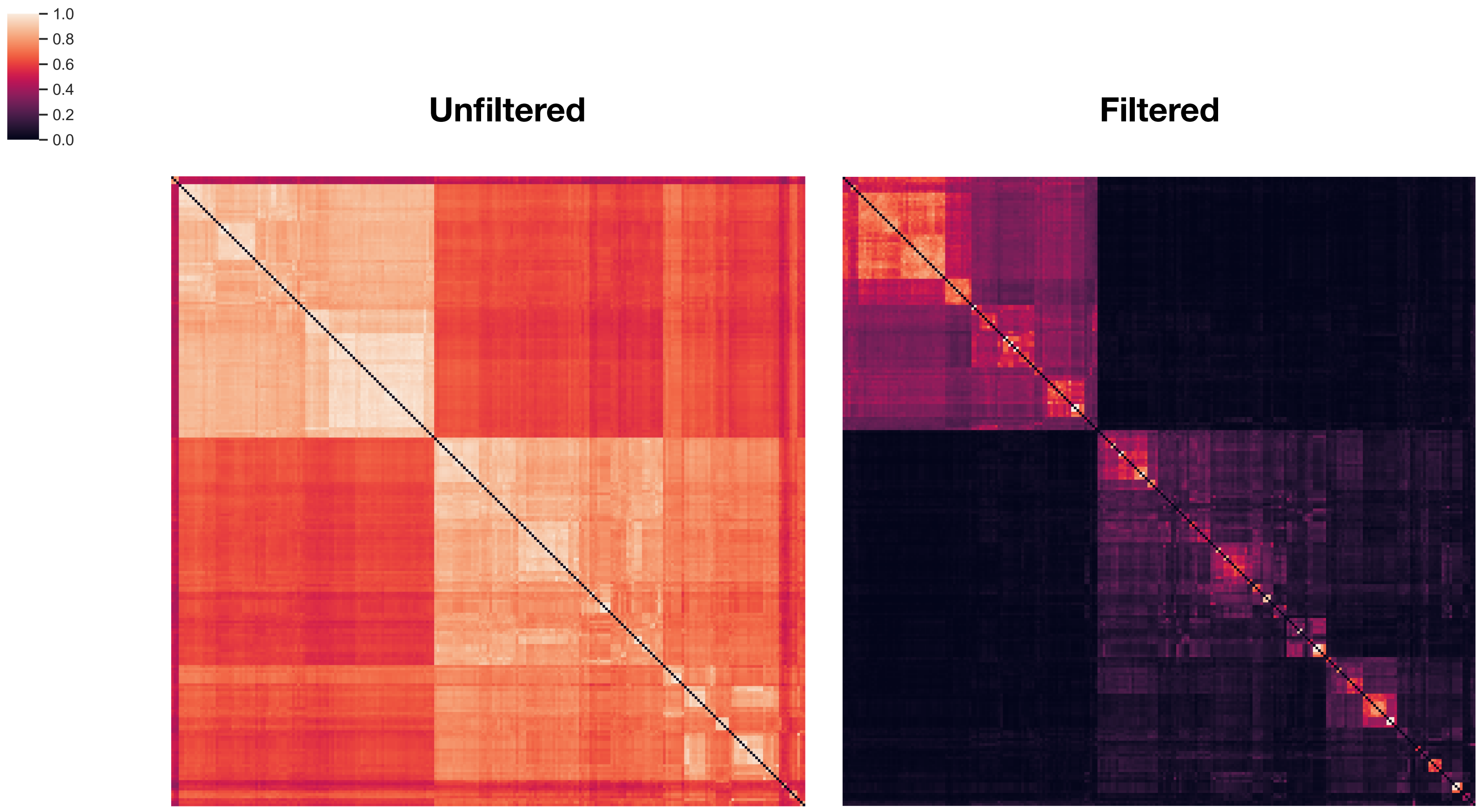


Figure 3.5: Heatmaps of coevolution similarity scores for the Trans241 dataset before and after filtering of common ECPs. The background level of coevolution similarity is significantly darker after filtering. However, enzyme pairs of high coevolution similarity i.e. bright spots in the heatmap, are still apparent.

This impact of the filtering can also be seen in Figure 3.5. The filtered heatmap is evidently of a far darker background, with an average similarity score of 0.11. However, bright spots of high similarity still exist, making it more obvious when two sequences have high coevolution similarity patterns that are more specific to individual EC classes.

As all four of the enzyme family datasets used in this work have diverse annotated functional subclasses, which make up at most 47% of a dataset (2.6.1.9 for the Trans986 dataset, Table 3.1), ECPs common to 60% of a dataset were filtered out all four datasets used in this work. While arbitrary, this threshold of 60% is likely high enough to remove family-wide coevolving pairs while not being so low that pairs common to entire functional subclasses are removed.

3.2.7 Enzyme substrate specificity prediction through label propagation

To test the predictive power of the networks with respect to the labelling of unannotated enzymes, and to analyse how well the networks captured the diversity in substrate specificity of the enzymes, a label-propagation experiment was performed using the EC labels of the datasets, using an algorithm inspired by the work of Schwikowski and coworkers [150], who performed label propagation on a yeast protein-protein interaction network. Starting with an initial representative subset of nodes that keep their labels, all other labels are removed. Based on the network structure, the algorithm then ”propagates” the labels of the initial subset and outputs EC numbers for the rest of the nodes over the course of the algorithm. The pseudocode for this algorithm is as follows (Algorithm 1):

```
Data: Nodes N, CanonicalLabels CL, MaxIterations I
Result: Dictionary of Predicted EC Labels PL
N, CL = initialization();
S, PL = iniRepSubset(CL);
N.remove(S);
count = 0;
while len(N) != 0 AND I != 100000 do
    node = sampleOne(N);
    neighbourLabels = countNeighbourLabels(node, S, PL);
    if neighbourLabels !=null then
        S += node;
        N.remove(node);
        PL[node] = pickTop2(neighbourLabels, M);
    end
    count +=1;
end
return PL;
```

Algorithm 1: Label propagation algorithm

The algorithm starts by initialising two variables: the list of nodes, N, of a network, either an SSN or a CSN, and a dictionary, CL, of the EC number of the enzyme taken from Swiss-Prot, with one or more EC numbers per enzyme in the network. The algorithm then outputs a dictionary PL containing the annotation predicted by the algorithm, where keys are entries and values are labels.

An initial random subset S of nodes is selected, and the dictionary PL with the labels for those nodes is initialised. S is a representative subset that contains at least one enzyme for each EC number. The algorithm then removes the nodes of S from N, and progressively predicts annotations for the rest of the nodes through label propagation.

After initialisation, the algorithm iterates through the remaining nodes in random order, ranking the labels of a node's neighbours based on the highest ranking EC numbers. If the node has annotated neighbours by that iteration, the node is added to S, is removed from N, and the top two most common EC numbers are assigned to the node in dictionary PL. This process is repeated until N reaches zero, or until a specified maximum number of iterations I is reached (in this case 100000). As this algorithm is stochastic, it is necessary to perform multiple rounds of label propagation.

In order to determine the predictive power of this approach when applied to SSNs and CSNs, the threshold was first iterated for both networks. This process allowed for the identification of an optimal threshold for each dataset for each network type for representing the available EC class annotations. With the thresholds optimised for this purpose, an in-depth comparison of how annotations propagate on the network structure could be made.

For each threshold from 0.05 to 0.9, 200 iterations of the label-propagation algorithm were performed. For each threshold three metrics were generated: average precision, average recall, and the F1 score, over the 200 iterations. For the purpose of this work, the assignment of predictions as true/false and positive/negative is based on whether a propagated EC class matches with the class that an enzyme node is annotated with. Therefore, for each node in a network, a true positive (TP) was when a correct class was propagated, a false positive (FP) when an incorrect class was propagated, and a false negative (FN) when no class was propagated. The precision and recall values were only calculated for nodes not included in the initial representative subset, and for nodes which have complete EC numbers. The equations for the precision, recall, and F1 score are the following:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

While a balance of both precision and recall is necessary to select the optimal threshold, more emphasis was given to recall when selecting the optimal threshold, in order to consider potential missing annotations and enzyme promiscuity. The optimal thresholds for SSNs and CSNs were identified by ranking the thresholds based on the F1 score, which is the harmonic mean of both the precision and the recall, and then selecting the threshold with the highest recall in the top 10 F1 scores.

3.2.8 Comparative analysis of SSNs and CSNs

Several metrics were computed in order to compare the topology of SSNs and CSNs, including the number of shared edges, the number of connected components of the network, and the number of edgeless nodes. The network comparisons were carried out at the optimal predictive threshold for both SSNs and CSNs, determined as described in section 3.2.7. The networks were visualized using the network visualization software Cytoscape [151].

Different connected components in a network can contain multiple protein clusters, comprised of enzymes with different substrate specificities. It is therefore important to know the distribution and variability of EC labels across clusters in the network. To achieve this quantification the MaxClust metric was computed. MaxClust is defined as the largest single connected component that contains nodes of a given EC number in a network. The size of the MaxClust for every EC label (the MCNumber) was first computed, followed by the fraction of nodes containing this EC label in the whole network that is covered by the MaxClust (MCFraction). From these values, a metric called the weighted-average MaxClust coverage (WAMCC) was computed (Equation 3.2). The WAMCC is calculated over an entire network, and represents how well enzymes of a similar substrate specificity are connected in a given network. While this metric does not provide details about individual specificity classes across components, it works well as an overall comparison metric to indicate the extent of enzyme substrate annotations that are grouped up across the network. Therefore, for the set of different EC classes E, with i an index over them, the WAMCC equation is formulated as follows:

$$WAMCC = \frac{\sum_{i=1}^{|E|} MCNumber_i * MCFraction_i}{NumNodes} \tag{3.2}$$

The number of EC numbers covered by each network was also computed; that is, how many of the EC classes have at least one node connected to at least one other node. All the metrics described in this section were computed for the optimised SSNs and CSNs, and also for the intersection network; the network produced by the set of edges that are shared by both an SSN and a CSN (SSN∩CSN).

3.2.9 Tertiary structure analyses of Trans986 and Croto99

The tertiary structure of enzymes is a better indication of functional conservation compared to primary sequence. Both the Trans986 and Croto99 datasets contained enough publicly available resolved three-dimensional structures that a structure-based analysis was able to be performed. The PDB structures, which number 32 for Trans986 and 18 for Croto99, were extracted. An all-vs-all comparison of their tertiary structures was then carried out using TM-align [80] for all pairs of nodes connected by an edge, for both the optimal SSN and CSN, for both of Trans986 and Croto99. Each alignment produced a TM-score between 0 and 1, with higher values indicating higher structural similarity.

The pattern of conservation of residues considered functionally important for both of these datasets was also analysed. This analysis was done by aligning sequences of interest using Clustal-Omega [67], so that the amino acid makeup of known functionally important positions could be investigated.

For the Trans986 dataset, five positions were of interest. Using the prephenate aminotransferase Q02635 as reference, these are Lys/Arg/Gln-12, Gly-39, Trp-125, Asn-175, and Arg-375 [152]. As for the Croto99 dataset, the type of reaction performed by a crotonase depends on the amount of negatively charged residues at three important positions [153]. For example, P76082, a hydratase (4.2.1.17), has the residues Glu-109, Glu-129, and Gly-137, two of which are negative residues, a pattern which is conserved across the hydratases of this family. The biochemical significance of these three positions was investigated in detail for the sequences in question.

3.3 Results

SSNs have been shown to be valuable for the functional analysis and annotation of enzymes and for the visual analysis of enzyme families. We therefore investigated the utility of CSNs for these applications in a comparative analysis with SSNs. First, the similarity of the network topology of between CSNs and SSNs was assessed, with an emphasis on how suitable their network structures are for the prediction of enzyme substrate specificity. Then, novel functional linkages revealed by the CSN were explored, particularly at low sequence identity thresholds.

3.3.1 Comparative analysis of CSN topology for the prediction of enzyme substrate specificity

As described in section 3.2.7, optimal thresholds for constructing SSNs and CSNs were chosen to provide a balance of precision and recall measures and were derived from label propagation experiments.

For the Trans241 dataset, the optimal thresholds were found to be 0.34 (or 34% sequence identity) for the SSN and 0.30 for the CSN. For the Trans986 dataset, the optimal thresholds were found to be 0.35 for the SSN and 0.52 for the CSN. For the Trans986 dataset, the optimal thresholds were found to be 0.35 for the SSN and 0.52 for the CSN. For the SDR142 dataset, the thresholds were 0.33 for the SSN and 0.45 for the CSN. Networks with these thresholds were considered to be optimally predictive for both families, based on currently available functional annotation.

The first row of Table 3.2 shows the number of edges in the CSN and SSN networks for the Trans241, Trans986, and SDR142 datasets. The optimal SSN networks had significantly more edges than the CSN, for all three datasets. The second row contains the number of connected components in each network. As edgeless nodes technically count as individual components, but are uninformative for this purpose, the edgeless nodes were subtracted. The resulting values are shown in parentheses. The last row of the Table 3.2 shows the number of edgeless nodes in each network. In each comparison the CSN also has fewer edgeless nodes than the SSN. As the optimal CSN has significantly fewer edges than the SSN, the fact that it also has fewer edgeless

Table 3.2: General network metrics for the optimal SSN and CSN for the Trans241, Trans986, and SDR142 datasests.

Dataset	**Trans241**		**Trans986**		**SDR142**	
Network (Threshold)	**SSN (0.34)**	**CSN (0.30)**	**SSN (0.35)**	**CSN (0.52)**	**SSN (0.33)**	**CSN (0.45)**
Edge Num	8,605	6,677	51043	43991	725	537
Component Num (- Edgeless)	11 (7)	6 (5)	26 (14)	18 (10)	41 (18)	38 (21)
Edgeless Node Num	4	1	12	8	23	17

nodes could demonstrate an increased robustness to change in thresholds.

Table 3.3 shows the distribution of EC coverage across the network components, as measured by the WAMCC value. The WAMCC value was computed for both the SSN and CSN. Table 3.3 also shows the number of EC labels covered by all of the networks and the intersection network. Disregarding edgeless nodes, both networks possess a similar number of components (Table 3.2), and a large overlap in the edges present exists as can be seen in the number of edges in the intersection network e.g. 94% of the Trans241 CSN edges exist in the SSN, and 73% of the Trans241 SSN edges exist in the CSN; 66% of the Trans986 CSN edges exist in the SSN, and 57% of the SSN edges exist in the CSN; 80% of the SDR142 CSN edges exist in the SSN, and 74% of the SSN edges exist in the CSN. This indicates that while the SSNs and CSNs may differ in detail, they do share a core network structure.

functional topology. For the Trans986 networks, the WAMCC values are similarly high, with values of 93.647 for the SSN and 93.865 for the CSN. The intersection network of this dataset also contains a significant amount of the total enzyme classes, with 24 of the 27 EC numbers being covered. Finally, for the SDR142 networks, the WAMCC values are similar: 53.283 for the SSN, and 54.434 for the CSN. Twenty-four of the 32 EC labels existed in the intersection network, showing that the level of agreement between the two networks encompass a large proportion of the functional diversity.

Table 3.3: Functional topology metrics for the optimal SSN and CSN for both the Trans241, Trans986, and SDR142 datasets. The metrics were also produced for the intersection of the two networks (SSN∩CSN).

	Trans241			**Trans986**			**SDR142**		
	Edges	**Covered ECs**	**WAMCC**	**Edges**	**Covered ECs**	**WAMCC**	**Edges**	**Covered ECs**	**WAMCC**
SSN	8,605	22	97.086	51043	24	93.647	725	24	53.283
CSN	6,677	25	99.174	43991	25	93.865	537	27	54.434
SSN∩CSN	6,343	18	-	29122	24	-	429	24	-

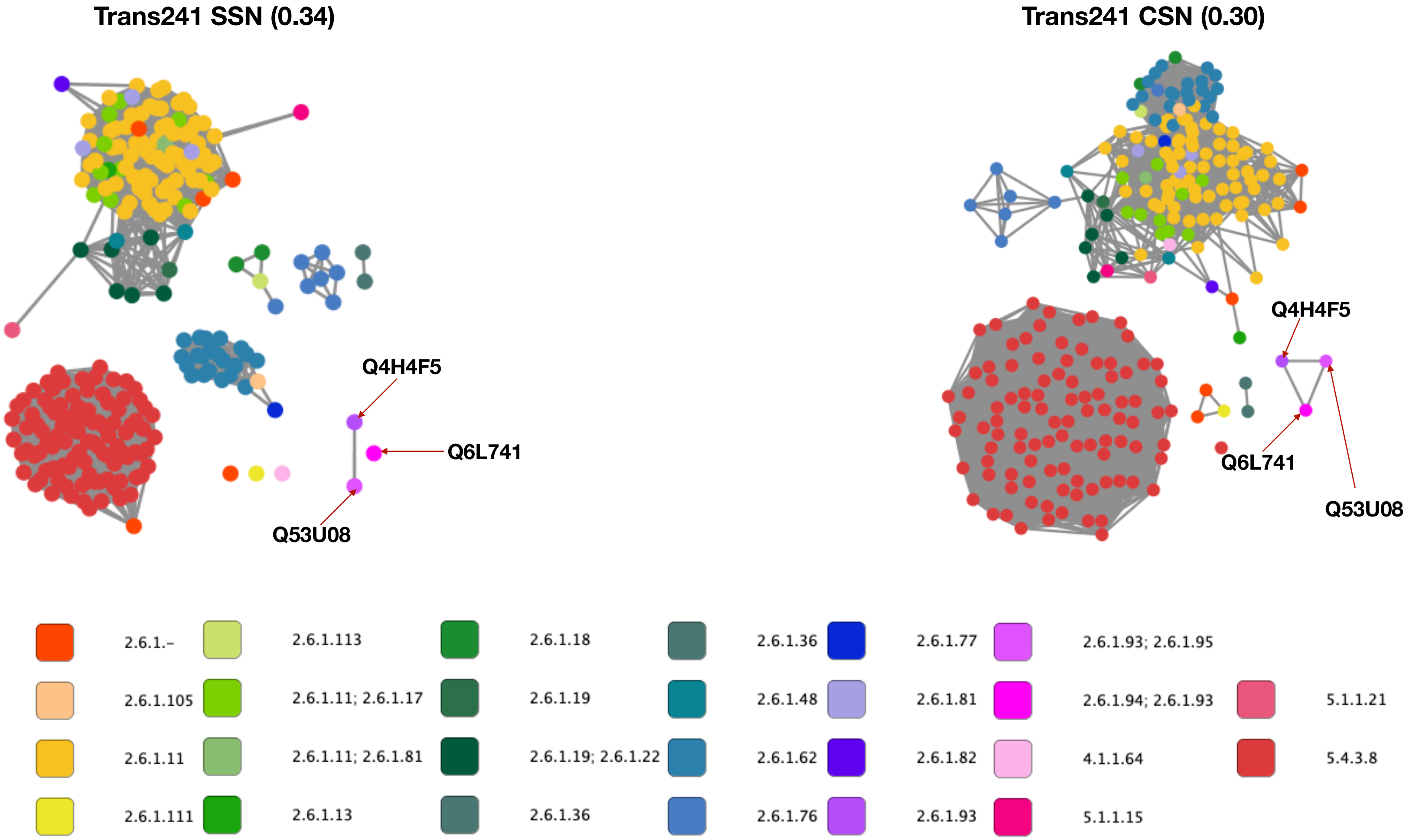


Figure 3.6: SSN and CSN at the chosen optimal thresholds for the Trans241 datasets. Nodes are coloured based on EC number label. Q4H4F5, Q53U08, and Q6L741 are all neamine transaminases (2.6.1.93). All three are fully connected in the CSN, while Q6L741 is edgeless in the SSN.

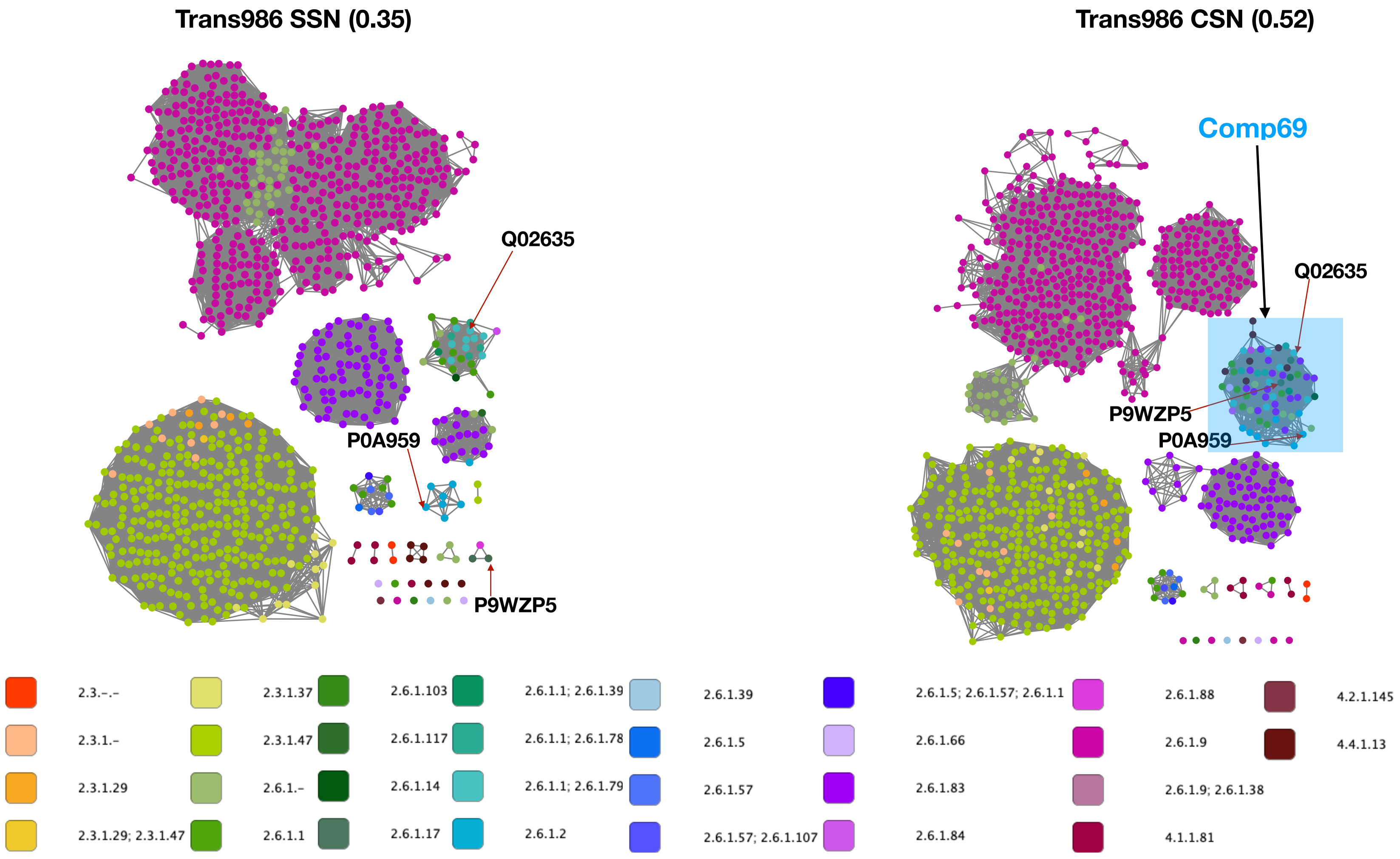


Figure 3.7: SSN and CSN at the chosen optimal thresholds for the Trans986 dataset. Nodes are coloured based on EC number label. Q02635 is a known prephenate aminotransefrase, and has high tertiary structure similarity to both P9WPZ5 and P0A959. This structural connection is picked up by the CSN with their inclusion into a single connected component, but not the SSN.

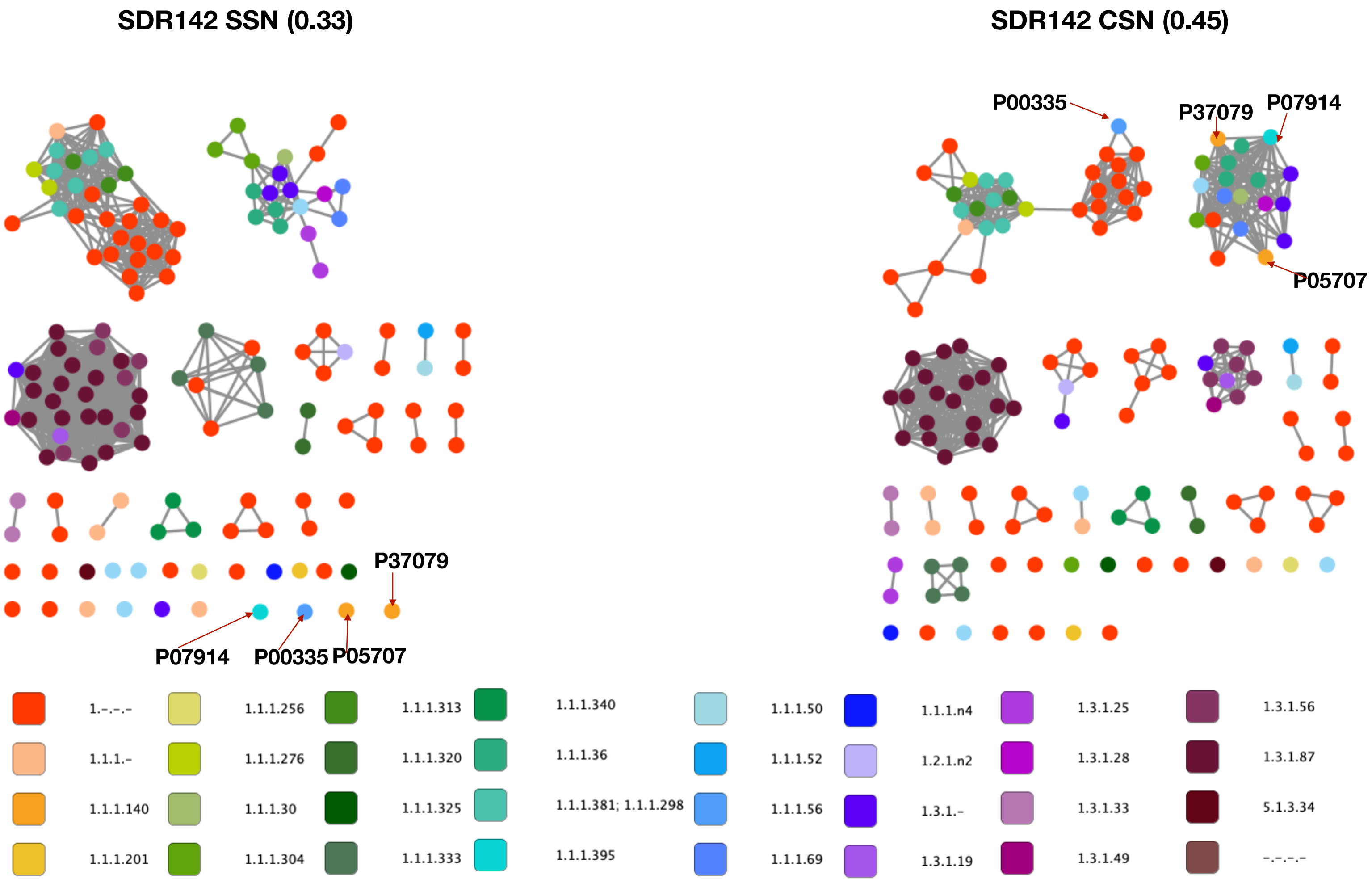


Figure 3.8: SSN and CSN at the chosen optimal thresholds for the SDR142 dataset. Nodes are coloured based on EC number label. P07914, P00335, P05707, and P37079 are four enzymes that represent three entire enzyme classes. The CSN connected them all, while the SSN left them all edgeless. Also, P00335 is connected to nodes with the incomplete EC number 1.-.-.-.

Table 3.4: Label Propagation experiment results for the optimal SSN and CSN for both the Trans241 and SDR142 datasets. To produce the precision and recall metrics, true positives are considered to occur when a correct annotation is predicted, false positives when an incorrect annotation is predicted, and false negatives when no annotation is predicted.

Dataset	**Trans241**		**Trans986**		**SDR142**	
Network (Threshold)	**SSN (0.34)**	**CSN (0.3)**	**SSN (0.35)**	**CSN (0.52)**	**SSN (0.33)**	**CSN (0.45)**
Precision	0.934	0.918	0.985	0.950	0.725	0.753
Recall	0.987	0.995	0.984	0.989	0.973	0.982

In order to compare the predictive value of SSNs and CSNs for annotation of enzyme substrate specificities, the label-propagation approach described in section 3.2.7 was applied to the networks to test the recovery of known annotations of the datasets under study. For the Trans241 dataset, the SSN showed slightly higher precision, and the CSN slightly higher recall (Table 3.4). While this result indicates that the two networks performed similarly in recovering the majority of the annotations, in some cases the SSNs failed to assign substrate specificities to several enzyme classes.

For example, of the three neamine transaminases (E.C. 2.6.1.93) in Trans241 - Q6L741, Q53U08, and Q4H4F5 - only the latter two were connected into a single component in the SSN, leaving one edgeless (Figure 3.6). Ideally, a functional network should connect all three together as they share the same enzymatic function, but the SSN fails to do this, leaving one of the nodes edgeless. In the CSN, however, all three were fully connected into a single component. Whilst the sequence similarity threshold could be lowered in order to add these connections to the SSN, it was necessary to lower the threshold to 29%, which caused an overall reduction in label propagation precision from 0.934 to 0.851. Label propagation tests the potential predictive power of these networks, and these results indicate that for the Trans241 dataset using the optimal CSN and SSN thresholds, the structure of the CSN could cover more of the substrate specificity distribution without sacrificing precision.

For the Trans986 dataset, the precision and recall values are similarly high for both the SSN and CSN (Figure 3.7). The SSN has a slightly higher precision, at 0.985, compared to the CSN which has a precision of 0.950. The recall is slightly higher for the CSN at 0.989, compared to the SSN with a recall of 0.984. Differences in precision is a harder metric to assess due to the possibility of missing annotation, but the significantly high values observed for both the SSN and CSN is evidence that the topologies of both network types are well suited for the purpose of recovering functional annotation of enzyme families.

For the SDR142 dataset, the optimal CSN had higher precision and recall than the SSN, as shown in Table 3.4. The CSN, which was smaller than the SSN for this family, included three more EC labels than the SSN (Table 3.3). These annotations are 1.1.1.140, 1.1.1.395, and 1.1.1.56, and apply to four proteins: P05707 and P37079; P07914, and P00335. These proteins are not connected by edges in the SSN (Figure 3.8), and were therefore not annotated during label propagation. Lowering the SSN threshold to 30% recovered the annotations of these enzymes, but doing so significantly lowered the precision, from 0.725 to 0.62. P00335 is connected to a cluster of nodes with the incomplete EC number 1.-.-.-, an observation which might imply that these proteins share a similar function to that of P00335. These data therefore show that the optimal CSN for the SDR142 dataset was able to connect a larger proportion of the enzymes in the class distribution than the SSN, without overly sacrificing precision.

3.3.2 The use of CSNs to make putative structural connections at low sequence identity

The tertiary structures were retrieved, where available on Swiss-Prot, for the Trans986 dataset, which numbered 32 structures. An all-vs-all comparison of their structures was performed using TM-align. A heatmap of the resulting TM-scores can be seen in Figure 3.9. This figure also shows the subgraph of these 32 sequences for both the optimal SSN and CSN.

In the CSN, there is a densely connected component that does not exist in the SSN (highlighted in blue in Figure 3.7). This component contains 69 nodes, and is therefore named Comp69 henceforth. When the 32 nodes with available tertiary structures are extracted into a subgraph, there are 25 edges between these nine Comp69 nodes of this subgraph in the CSN, while there are only 7 in the SSN. From looking at the position of these sequences in TM-score heatmap, it is clear that they were clustered together as a clade in the dendrogram, and that indeed they have higher TM-scores within this connected component compared to scores between other groups.

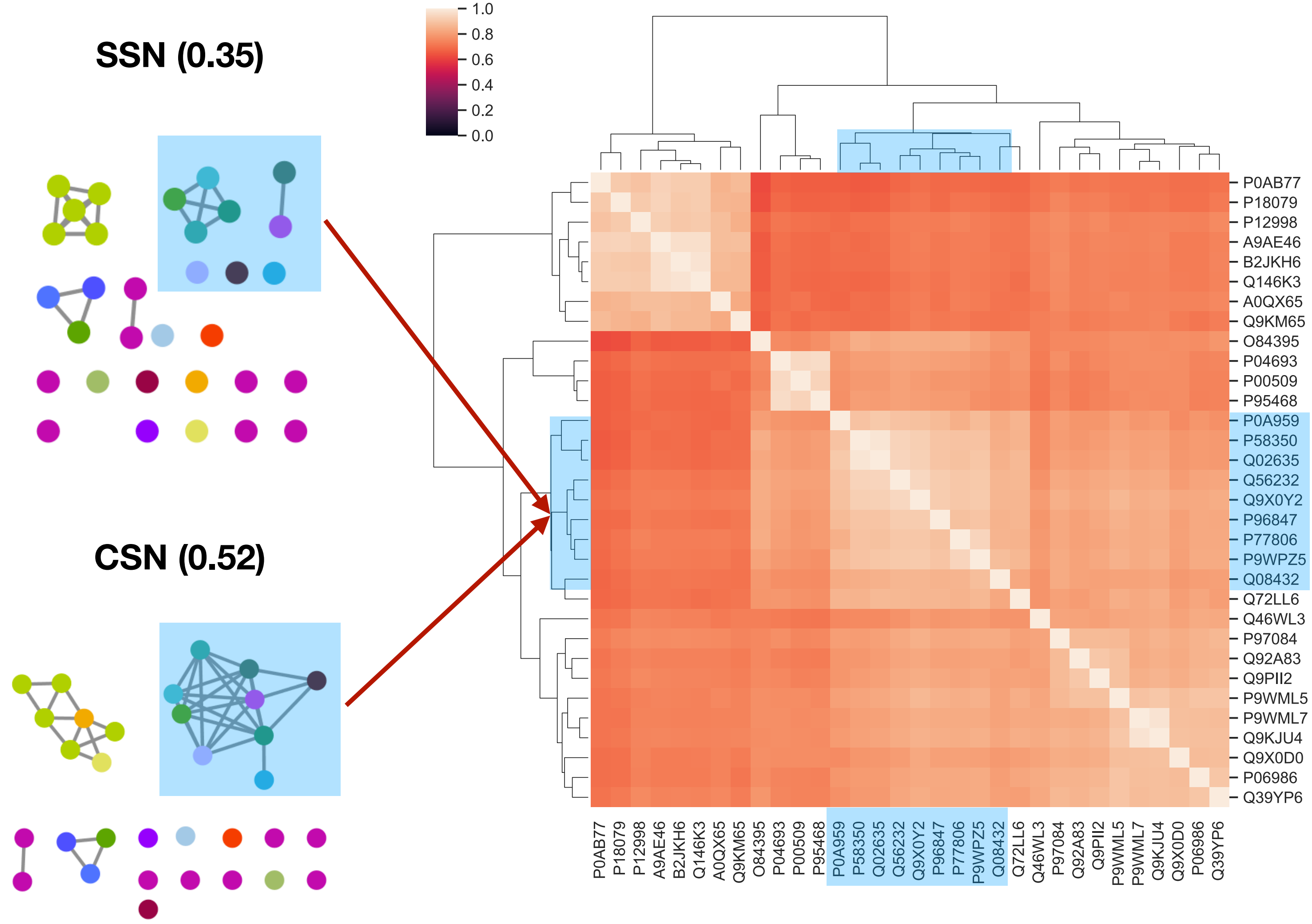


Figure 3.9: Heatmap of the resulting TM-scores for the 32 tertiary structures of Trans986 after an all-vs-all tertiary structure alignment using TM-align. The brighter the colour, the more similar two tertiary structures are. The sequences in the clade highlighted in blue on the heatmap represent a densely connected component in the subgraph of these sequences in the Trans986 CSN. These same sequences are vastly less connected in the SSN.

Many of the 25 edges are not present in the SSN; this fact implies that the CSN has connected sequences with relatively low sequence identity but high similarity in tertiary structure. Evidence for this observation can be seen in Figure 3.10, with the superposition of the structures of P9WPZ5 and Q02635 produced by TM-align. These two sequences share only 29.4% sequence identity, and yet have a high TM-score of 0.899. These structures are therefore very similar, especially relative to their sequence identity. When the same comparison is performed between P0A959 and Q02635 the effect of tertiary similarity, and low sequence identity, is more pronounced: the sequence identity is even lower at 26.6%, with the TM-score being slightly higher at 0.909.

The same analysis was performed on the Croto99 dataset, which had 18 sequences with available tertiary structures. Q5LLW6 is a methylthioacryloyl-CoA hydratase (4.2.1.155) that is left edgeless in the SSN until around 30% sequence identity (Figure 3.11). The CSN however, places it in a cluster with other hydratases, mainly enoyl-CoA hydratases like P76082, but also crotonyl-CoA hydratases like P52046. All three of these enzymes have tertiary structures available, and Q5LLW6 has TM-scores of 0.9 when compared with P52046 and 0.86 when compared with P76082. The sequence identity of Q5LLW6 with these enzymes is 29.10% for P52046 and 30% for P76082.

Given that much of the power of coevolving residues lies in their strong correlation with residues being in physical contact, these results provide evidence that CSNs are able to connect enzymes of low sequence identity but high structural similarity. This means that not only are CSNs sensitive to structurally similar proteins, it is likely this sensitivity comes from the CSN using the inherent structural knowledge contained in coevolving residues when making its connections. Given the importance of tertiary structure for the function of enzymes, the ability to make structural connections at the twilight zone of sequence identity, it is proposed that the use of CSNs for enzyme family analysis has the potential for the discovery of similarly functioning enzymes that are further apart on the evolutionary scale.

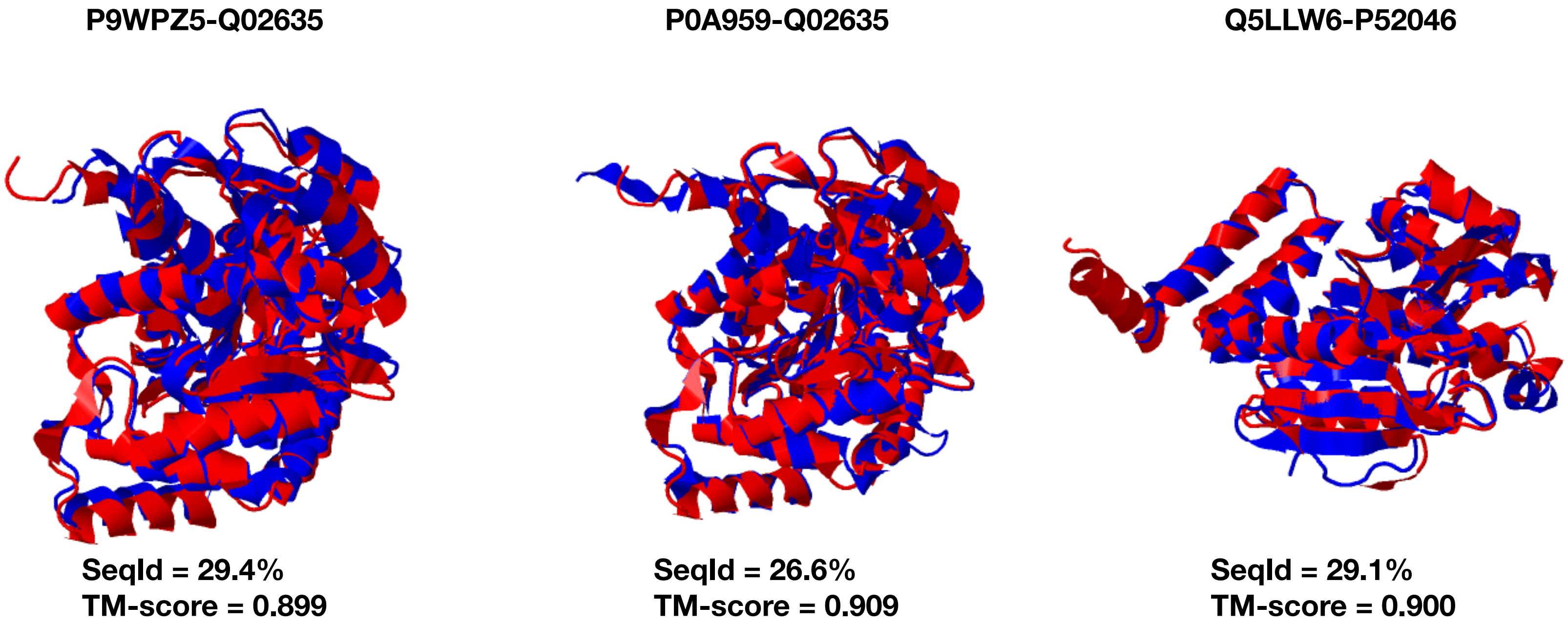


Figure 3.10: The resulting superposition of tertiary structures for P9WPZ5-Q02635, P0A959-Q02635 in the Trans986 dataset, and Q5LLW6-P52046 in the Croto99 dataset, produced by TM-align. All three of these tertiary structure alignments are at relatively low sequence identities, and yet have high tertiary structure similarity. All three of these structural relationships were retrieved by the CSN, but not the SSN.

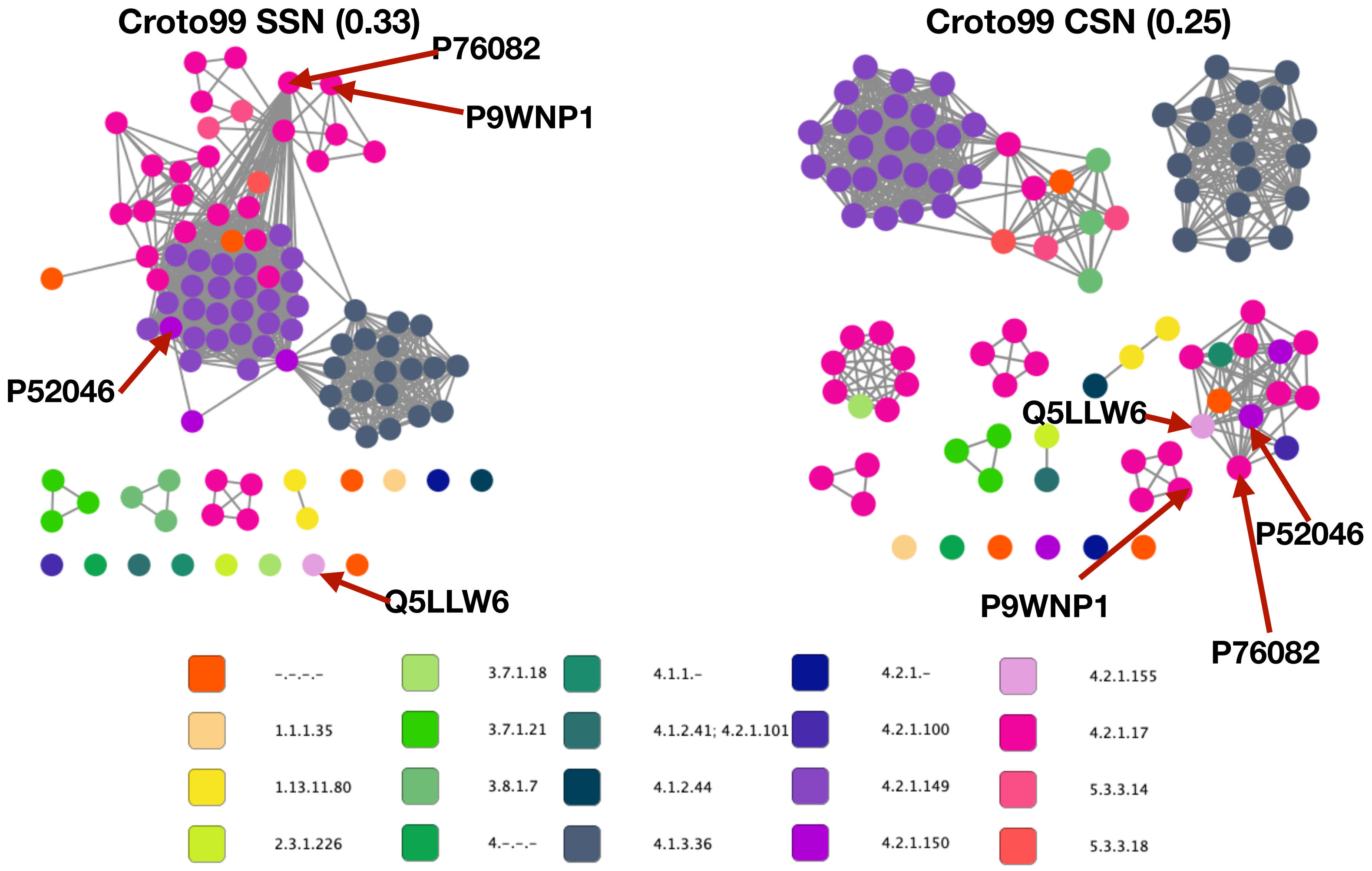
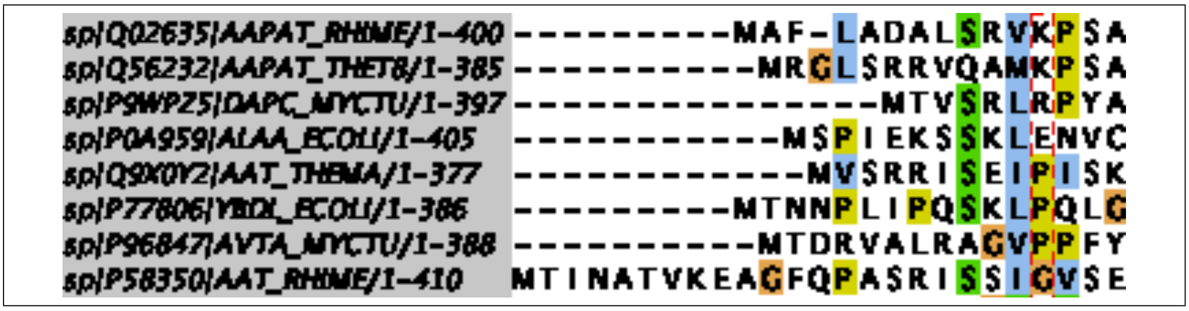


Figure 3.11: SSN and CSN at the chosen optimal thresholds for the dataset made up of 99 crotonases, Croto99. Nodes are coloured based on EC number label. P76082 and P9WNP1 are two of the 27 nodes labelled as 4.2.1.17 (enoyl-CoA hydratase). The SSN connected all of these nodes in such a way that the label propagation algorithm labels them with 4.2.1.17, while the CSN split them in separate groups.



```
sp|Q02635|AAPAT_RHIME/1-400 ---------MAF-LADALSRVKPSA
sp|Q56232|AAPAT_THET8/1-385 ----------MRGLSRRVQAMKPSA
sp|P9WPZ5|DAPC_MYCTU/1-397  ---------------MTVSRLRPYA
sp|P0A959|ALAA_ECOLI/1-405  -----------MSPIEKSSKLENVC
sp|Q9X0Y2|AAT_THEMA/1-377   ------------MVSRRISEIPISK
sp|P77806|YBDL_ECOLI/1-386  ---------MTNNPLIPQSKLPQLG
sp|P96847|AVTA_MYCTU/1-388  ----------MTDRVALRAGVPPFY
sp|P58350|AAT_RHIME/1-410   MTINATVKEAGFQPASRISSIGVSE
```

Figure 3.12: A segment of a multiple sequence alignment of eight Trans986 enzymes connected by the CSN into a connected component (Comp69), with the Lys-12 position highlighted (using Q02635 as reference). Lys-12 is an important residue for prephenate aminotransferase function, and has been shown to conservatively be substituted by Arn or Gln.

3.3.3 The use of CSNs to indicate putative functional connections at low sequence identity

It was therefore of interest to try and find indications of functional similarity among these structural connections. In the Trans986 dataset, Q56232 and Q02635 were both confirmed to be prephenate aminotransferases (2.6.1.78 and 2.6.1.79, respectively), with a significantly similar tertiary structure (TM-score of 0.941). As shown previously, both P0A959 and P9WPZ5 showed a high structural similarity to Q02635 (Figure 3.10), and are both in the Comp69 component (Figure 3.9). According to both Swiss-Prot and the literature, there are five functionally important residues for prephenate aminotransferase activity [152]. Taking Q02635 as reference, these are: Lys-12, Gly-39, Trp-125, Asn-175, and Arg0375. Lys-12 is a particularly important residue for this substrate, and has also been substituted by Arg or Gln in known prephenate aminotransferases. Eight of the nine sequences of the Comp69 nodes with available tertiary structures have at least the latter four resides conserved, with varying residues at the Lys-12 position. Figure 3.12 shows an MSA of these eight sequences at the Lys-12 position.

This alignment shows that P9WPZ5 does indeed have an Arg at that position. These similarities at functionally important residues, combined with the overall tertiary structure similarity to known prephenate aminotransferases, might indicate that P9WPZ5 could also share prephenate aminotransferase activity. It would be interesting to confirm the activity of P9WPZ5 in the laboratory, as currently its only known function is

Table 3.5: Difference in residue conservation values at the five positions important for prephenate aminotransferase function (using Q02635 as reference) between the Comp69 sequences of Trans986 and the dataset as a whole. There are several major differences in residue conservation between these two groupings.

	Lys-12	**Gly-39**	**Trp-125**	**Asn-175**	**Arg-375**
Comp69	Pro (35%) Lys (19%) Thr (10%)	Gly (90%) Ala (10%)	Tyr (61%) Trp (35%) Phe (4%)	Asn (100%)	Arg (100%)
All	Lys (17%) Arg (13%) Thr (10%)	Asn (80%) Gly (16%) Met (2.8%)	Tyr (34.7%) Phe (39.1%) Asp (22.9%)	Asn (59.1%) Ser (34%) Ala (6.5%)	Arg (100%)

as a succinyldiaminopimelate transaminase (2.6.1.17).

P0A959 has the same residues at these positions as P9WPZ5 except for the important Lys-12 position, where it has a Glu (Figure 3.12). Glu has not yet been confirmed as a residue that grants prephenate aminotransferase function, however some known enzymes of this function do have a Gln instead [152]. It is known that a Gln-Glu substitution is often conservative, so there is a possibility that Glu might fulfil the same role in the catalysis of the reaction. This possibility is further enhanced by the strong tertiary structure similarity between P0A959 and other known prephrenate aminotransferases like Q02635 (Figure 3.10).

All of the other Comp69 sequences have residues at Lys-12 that make them unlikely to have prephrenate substrate affinity, as three of them contain a Pro which is generally unreactive, and the final one contains a Glycine which has been shown to make these enzymes unable to perform this reaction [152]. However, given the clear connection in terms of tertiary structure similarity, there is even the possibility of currently unannotated functions that some these sequences might share currently unknown enzymatic functions.

Further indications of this possibly new annotation are provided by considering the residue makeup at the five functionally-important positions for the sequences within the entire Comp69 component compared to the rest of the Trans986 dataset. An MSA was generated for the Comp69 sequences to examine the conservation patterns for the five functionally important residues. This resulted in the residue distributions shown

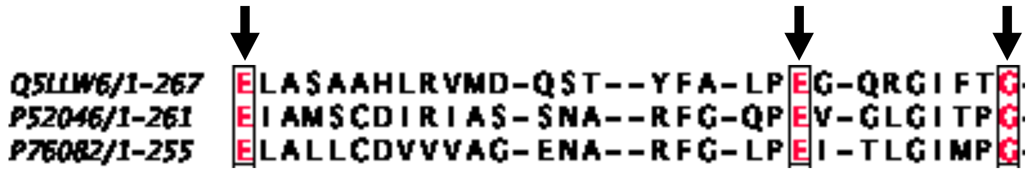

Q5LLW6/1-267 ELASAAHLRVMD-QST--YFA-LPEG-QRGIFTG
P52046/1-261 EIAMSCDIRIAS-SNA--RFG-QPEV-GLGITPG
P76082/1-255 ELALLCDVVVAG-ENA--RFG-LPEI-TLGIMPG

Figure 3.13: Multiple sequence alignment (MSA) of three nodes in Croto99 that are known hydratases - Q5LLW6 is a methylthioacryloyl-CoA hydratase, P76082 is a enoyl-CoA hydratase, and P52046 is crotonyl-CoA hydratase. The CSN connected all three of these nodes, indicating a potential sharing of substrate function. The three positions of the MSA pointed at with arrows are the three key residues that decide the type of reaction a crotonase catalyses. For hydratases, two negatively charged residues, usually glutamates, are necessary.

in Table 3.5.

There is a clear difference in the residue conservation at these positions between the Comp69 sequences and the total Trans986 dataset, with Arg-375 being the only one that is 100% conserved in both. At Lys-12, while the proportion of Lys and Thr is similar in both sets, Pro makes up a significant proportion of the residue types at this position, with over a third of the component sequences having a proline compared to just 8.8% of the dataset as a whole. The difference at Gly-39 is even more stark - while 90% of the component sequences have a glycine, only 16% of the total datset does, with asparagine instead being the dominant conserved residue with 80% conservation. Trp-125 has a strong aromatic hydrophobic residue concentration in both sets, although there is a clear preference for tyrosine in the component sequences. Finally, while the proportion of asparagine is high at Asn-175 in both sets, it is universal to the Comp69 sequences while only 59.1% of the total dataset have an asparagine. While assertions of functional similarity are not certain without experimental evidence, these large differences in residue makeup between the Comp69 enzymes and the full dataset are strong pointers of such assertions.

More evidence of functional connections at low identity being made by the CSN exists in the Croto99 dataset. As mentioned earlier, the Croto99 CSN was sensitive enough to recover low sequence identity structural relationships between Q5LLW6 and other crotonases like P52046 and P76082 (Figures 3.10, 3.11). The conservation of the three functionally important residues reported by Grishin *et al* (2012), was observed and shown in Figure 3.13. All three of these sequences have the residues characteristic of hydratases at the three functionally important positions: Glu, Glu, and Gly, reinforcing the hypothesis that they may be functionally similar enzymes. This linkage is confirmed by the literature [154], as Q5LLW6 does indeed catalyse the hydration of crotonyl-CoA as a secondary substrate, P52046's primary substrate. This functional linkage is evidence of the CSN connecting enzymes of low sequence similarity that share not just tertiary structure, but also catalytic function.

Given that Q5LLW6 and P52046 share only 29.10% sequence identity, have a high TM-score of 0.9, (Figure 3.10) and there is experimental evidence supporting this functional similarity, these results further illustrate the value of CSNs for establishing putative functional similarity between two enzymes of low sequence identity, down to the substrate specificity level.

3.3.4 The identification of putative evolutionary connections at low sequence identity using CSNs

A phylogenetic tree was produced for the Trans986 dataset (Figure 3.14), with an emphasis on the Comp69 component. The trees were generated with default parameters using FastTree [155], which produces approximately-maximum-likelihood phylogenetic trees. The tree was then visualised using IcyTree [156]. The structure of the phylogenetic tree was compared to the layout of the Comp69 sequences on the optimal SSN, which is split into 5 different connected components (Figure 3.14). These components are referred to as C1, C2, C3, C4, and C5. It is also made up of six edgeless nodes, which are referred to as NC (no-component).

The Comp69 sequences are very close to each other on the tree, especially relative to the evolutionary timeline of the total dataset, and they are separated into five recognisable subclades - the same number of components in the SSN. Indeed, the C2, C3, C4, and C5 clades all match exactly with their respective components in the SSN, while the C1 component has just one node outside of the clade all others belong to. The edgeless nodes are spread all around this subtree, but are still located overall in this space of the tree, with at least three of them forming their own clade next to the C4 clade.

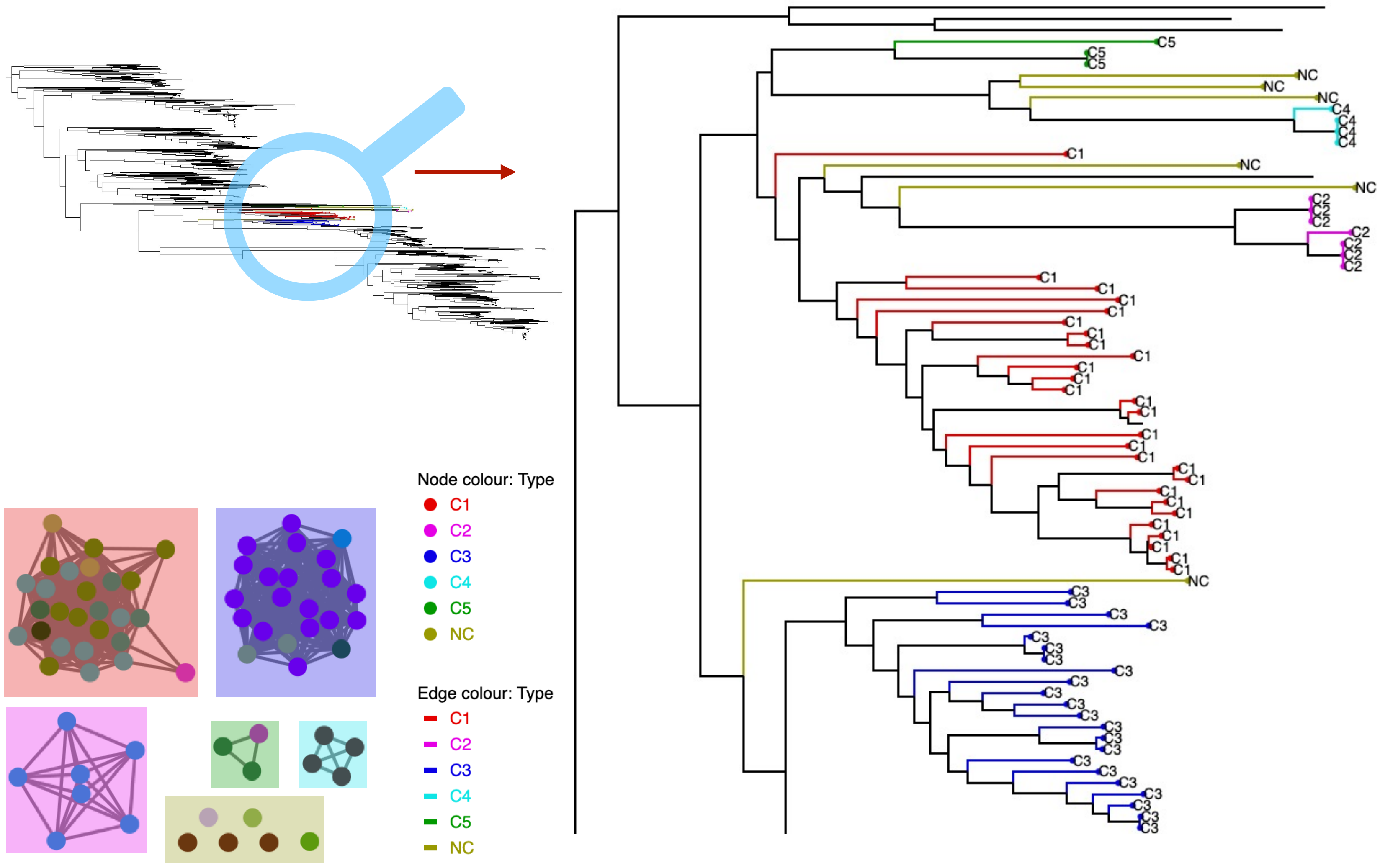


Figure 3.14: Phylogenetic analysis of the Comp69 component from the optimal CSN of the Trans986 dataset. The Comp69 sequences are all part of the same major clade, which can be separated into five major sub-clades, showing a relatively significant phylogenetic relationship between the Comp69 sequences. These five sub-clades correspond in clustering to the five connected components the optimal SSN separates Comp69 into. However, Comp69 contains six nodes that are left edgeless in the optimal SSN, which are spread around the major clade corresponding to the Comp69 sequences. This inclusion of six nodes that are left edgeless in the SSN into a major phylogenetic clade shows the ability of CSNs to group up phylogenetically similar sequences at low sequence identity thresholds.

The Comp69 sequences are therefore clearly also close phylogenetically. Combined with the proven tertiary structure similarity and the interesting aforementioned residue frequency at important positions, there is strong evidence that the CSN connecting all of these 69 sequences into a single densely connected component has confident validity and is likely worth further investigation experimentally. It is clear that the CSN is able, in this example, to detect sensitive relationships between sequences that are structural, functional, and even evolutionary in nature, than the SSN.

3.3.5 The use of CSNs to reveal inconsistencies in functional annotation

The optimization of SSN and CSN thresholds for the Croto99 dataset and its annotations generated networks in which the SSN significantly outperformed the CSN for both precision and recall. For example, at 33% identity, the SSN had a precision of 0.964 and a recall of 0.973 after label propagation. The CSN at a threshold of 0.25, however, had a precision of 0.883 and a recall of 0.902; largely worse results than the SSN based on these metrics.

Further investigation showed that the CSN could not meaningfully cluster the 27 nodes labelled as enoyl-CoA hydratase (4.2.1.17) at any threshold, but separated them into multiple different clusters. In contrast, the SSN could reliably propagate the 4.2.1.17 label to these nodes (Figure 3.11).

Because this method of threshold optimisation depends on accurate annotation, further investigation on the level of evidence supporting the annotation of these 27 enzymes was performed. On Swiss-Prot, the annotation of only two of these enzymes is supported by experimental evidence, while the other 25 are annotated based on similarity, and have the protein name ”probable enoyl-CoA hydratase”, meaning this annotation might be unreliable.

The possibility that the annotation of these enzymes could be incorrect was therefore explored. As mentioned in section 3.2.9, the tertiary structures of 18 Croto99 enzymes were retrieved for an all-vs-all comparison. While the enzymes represented in Croto99's supposedly optimal SSN share a high degree of tertiary structure similarity, as indicated by their comparative TM-scores, having a median of 0.92, there

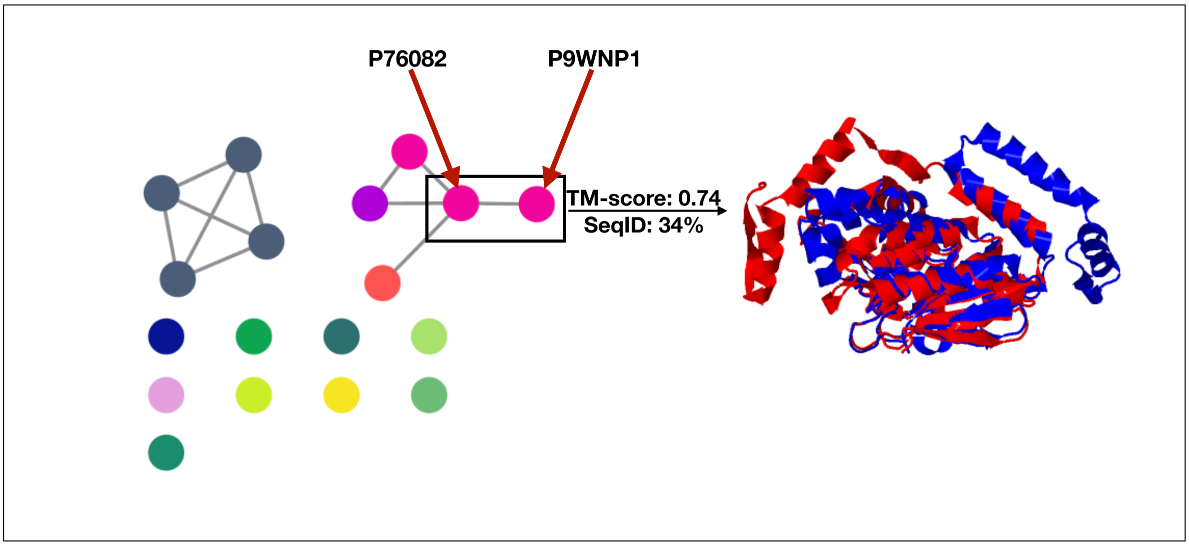
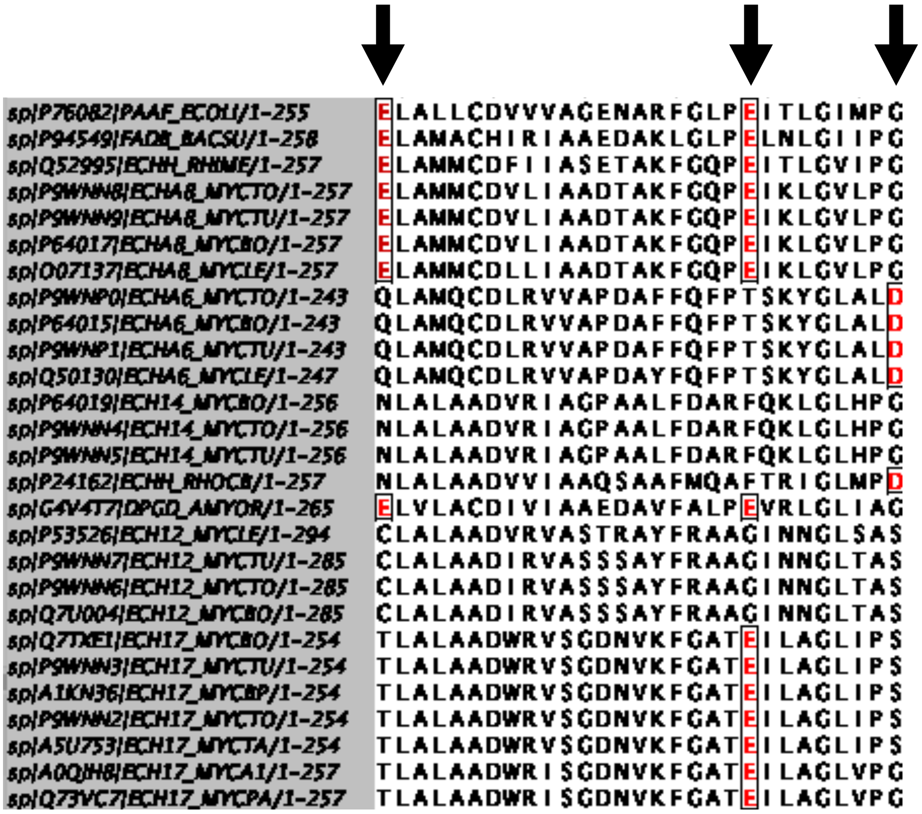


Figure 3.15: SSN at 33% identity for the 18 sequences of Croto99 that have tertiary structures available. Nodes are coloured based on EC number. The nodes in the black box represent P76082 and P9WNP1. The overlapping of their tertiary structures can be seen on the right.

was one outlier edge with a TM-score under 0.75 between P76082 and P9WNP1, both of which are reportedly enoyl-CoA hydratases (4.2.1.17) (Figure 3.15). These two enzymes share 34% sequence identity, and yet, as is evident in Figure 3.15, there is a significant difference in the orientation of multiple secondary structures, which reduces the confidence in some of the functional linkages of this SSN.

Since the type of reaction undergone by a crotonase depends on the number of negatively charged residues in three specific positions, it would be expected that all 27 of these putative hydratases would have two negatively charged residues [153], which was examined by looking at an alignment of the sequences. It is apparent that this required biochemical profile for a hydratase is not present for 19 of the sequences (Figure 3.16).

For example, P9WNP1 has only one negatively-charged residue at any of these positions, an aspartate at the last position. Others, like P9WNN5 and Q7U004, do not have any negatively-charged residues at those positions. This result puts into doubt many of the 4.2.1.17 annotations in Croto99, and was made possible through the addition of a complementary view of the dataset by using CSNs. CSNs could therefore also be used for the verification of currently existing annotations by either confirming existing linkages revealed with sequence similarity, or by pointing out potential inconsistencies.



```
sp|P76082|PAAF_ECOLI/1-255       ELALLCDVVVAGENARFGLPEITLGIMPG
sp|P94549|FADB_BACSU/1-258       ELAMACHIRIAAEDAKLGLPELNLGIIPG
sp|Q52995|ECHH_RHIME/1-257       ELAMMCDFIIASETAKFGQPEITLGVIPG
sp|P9WNN8|ECHA8_MYCTO/1-257      ELAMMCDVLIAADTAKFGQPEIKLGVLPG
sp|P9WNN9|ECHA8_MYCTU/1-257      ELAMMCDVLIAADTAKFGQPEIKLGVLPG
sp|P64017|ECHA8_MYCBO/1-257      ELAMMCDVLIAADTAKFGQPEIKLGVLPG
sp|O07137|ECHA8_MYCLE/1-257      ELAMMCDLLIAADTAKFGQPEIKLGVLPG
sp|P9WNP0|ECHA6_MYCTO/1-243      QLAMQCDLRVVAPDAFFQFPTSKYGLALD
sp|P64015|ECHA6_MYCBO/1-243      QLAMQCDLRVVAPDAFFQFPTSKYGLALD
sp|P9WNP1|ECHA6_MYCTU/1-243      QLAMQCDLRVVAPDAFFQFPTSKYGLALD
sp|Q50130|ECHA6_MYCLE/1-247      QLAMQCDLRVVAPDAYFQFPTSKYGLALD
sp|P64019|ECH14_MYCBO/1-256      NLALAADVRIAGPAALFDARFQKLGLHPG
sp|P9WNN4|ECH14_MYCTO/1-256      NLALAADVRIAGPAALFDARFQKLGLHPG
sp|P9WNN5|ECH14_MYCTU/1-256      NLALAADVRIAGPAALFDARFQKLGLHPG
sp|P24162|ECHH_RHOCB/1-257       NLALAADVVIAAQSAAFMQAFTRIGLMPD
sp|G4V4T7|DPGD_AMYOR/1-265       ELVLACDIVIAAEDAVFALPEVRLGLIAG
sp|P53526|ECH12_MYCLE/1-294      CLALAADVRVASTRAYFRAAGINNGLSAS
sp|P9WNN7|ECH12_MYCTU/1-285      CLALAADIRVASSSAYFRAAGINNGLTAS
sp|P9WNN6|ECH12_MYCTO/1-285      CLALAADIRVASSSAYFRAAGINNGLTAS
sp|Q7U004|ECH12_MYCBO/1-285      CLALAADIRVASSSAYFRAAGINNGLTAS
sp|Q7TXE1|ECH17_MYCBO/1-254      TLALAADWRVSGDNVKFGATEILAGLIPS
sp|P9WNN3|ECH17_MYCTU/1-254      TLALAADWRVSGDNVKFGATEILAGLIPS
sp|A1KN36|ECH17_MYCBP/1-254      TLALAADWRVSGDNVKFGATEILAGLIPS
sp|P9WNN2|ECH17_MYCTO/1-254      TLALAADWRVSGDNVKFGATEILAGLIPS
sp|A5U753|ECH17_MYCTA/1-254      TLALAADWRVSGDNVKFGATEILAGLIPS
sp|A0QJH8|ECH17_MYCA1/1-257      TLALAADWRISGDNVKFGATEILAGLVPG
sp|Q73VC7|ECH17_MYCPA/1-257      TLALAADWRISGDNVKFGATEILAGLVPG
```

Figure 3.16: Multiple sequence alignment (MSA) of the 27 nodes in Croto99 labelled as 4.2.1.17 (enoyl-CoA hydratase). The three positions of the MSA indicated are the three key residues that decide the type of reaction a crotonase catalyses. For hydratases, two negatively charged residues, usually glutamates, are necessary. Only 8 out of the 27 sequences satisfy this requirement, even though they are all labeled as hydratases.

This complementarity applies in both directions, with SSNs also being useful for confirming assertions made by CSNs . For example, in the CSN for Croto99, Q9XB60 and O69762 were connected, and yet their TM-score was 0.79 with a heavy mismatch in active site biochemistry. These two sequences are not connected in the SSN, as their sequence identity is 22%. This finding further reinforces the idea that the use of the two complementary methods can help provide a better understanding of a dataset, rather than considering the approaches as two competing methods.

3.4 Discussion and conclusions

With the rapidly increasing number of protein sequences in public repositories, new approaches for analysing and annotating protein families at increased scale are important to fully understand the functional and evolutionary aspects of enzymes. Such approaches would be especially helpful for the selection process of enzymes to be characterised *in vitro*. To this end, SSNs are increasingly being used for the inspection and functional analysis of protein families. Whilst not as powerful as a phylogenetic analysis for elucidating the track of evolution, the depiction of the functional relationships between proteins in a family as a network allows the visualisation and analysis of trends and groupings within large families [74, 136]. Representation of protein families as networks also makes their data accessible to common graph-based analytical approaches, metrics and tools such as cluster analysis. SSNs have also been applied to the functional analysis of enzyme protein families, where they can help resolve substructure in a family and be used to assert functional equivalence. Whilst useful, SSNs have limitations related to the use of sequence identity as proxy for homology, as described in section 2.2.3. There is therefore a need for alternate approaches for the construction of protein similarity networks.

In this work, coevolution similarity networks (CSNs) are proposed as a new approach for building networks of enzyme protein families, that can help reveal family structure and functional relationships, but are not based directly on sequence similarity. As residues that coevolve do so under evolutionary pressure to maintain the stability of protein structure and function [145], similarity networks built out of a family-wide comparison of coevolution patterns are likely to display the important structural and functional relationships for that family. Specifically, the applicability of CSNs down to the level of substrate specificity was explored, with a particular emphasis on linkages existing at low sequence identity thresholds.

3.4.1 Strengths and limitations of CSNs

The results presented here show that CSNs offer another useful approach for the analysis and annotation of large protein families. CSNs and SSNs generally agree on the overall network and functional topology of the protein families. The majority of enzyme classes are distributed similarly in both types of network, and both network types performed similarly in the reassignment of annotation to nodes in the label propagation experiments. However, in this work CSNs were shown, in some cases, to be able to reveal interesting linkages that exist at lower sequence identity thresholds. CSNs were able to connect enzymes with highly similar tertiary structures yet sequence identities deep into the twilight zone of homology, down to 26.5% identity (Figure 3.10), hinting at possible shared function. These revealed connections at lower identity thresholds are likely due to the inherent structural information contained in residue-residue coevolution data.

It was also shown that many of these structural connections exhibit residue conservation at positions known to be functionally important at an enzyme subclass level (Figures 3.12-3.13). These connections were also shown to be between phylogenetically close enzymes, despite the lower sequence identity (Figure 3.14). Complementing these features, CSNs were shown to also help identify clear discrepancies and formulate hypotheses about the correctness of existing enzyme annotation (Figures 3.15- 3.16).

Finally, these described strengths of the CSN method are also valuable in the context of a lack of dependence on sequence identity as the primary metric used to group up enzymes of the same putative catalytic function. As was described in section 2.2.3 of the Background, this dependence is a known limitation of current approaches for generating diverse enzyme panels, and approaches tackling it are therefore valuable. CSNs provide a promising contribution that was shown in this chapter to provide credible hypotheses of functional groupings in an enzyme family without a reliance on sequence identity, but rather through a comparison of residue-residue coevolution patterns.

A disadvantage of CSNs is that they are less easily constructed than SSNs; they are computationally more complex to produce, and coevolution is less intuitive to interpret directly than sequence similarity. Also, CSN network construction currently uses arbitrary values for two parameters - the size N of residue-residue coevolution networks, and the filter F that discards coevolution pairs common to a dataset - these parameters could be optimised and maybe even improve the performance of CSNs.

Also, much like the SSN, edges made in the CSN will not always be accurate for the purpose of predicting enzyme function down to substrate specificity, as shown in section 3.3.5.

Another limitation of this work is that unlike SSNs, CSN thresholds are not consistent in terms of significance. Sequence identity thresholds, while rife with exceptions, are easier to formulate correlations and rules about the likelihood of homology at certain thresholds, with 40% often being used as an arbitrary but provenly successful guideline. Indeed, the optimal thresholds used in this work for the Trans241, Trans986, and SDR142 datasets were 0.34, 0.35, and 0.33, respectively. For the CSN however, the optimal thresholds were 0.3, 0.52, and 0.45 for the same datasets respectively. The reason for this difference is unclear, but is likely to be related to the higher levels of parametrisation of CSNs compared to SSNs, in particular due to the use of unoptimised values for said parameters.

3.4.2 A de novo complementary similarity network approach to the functional analysis of diverse enzyme families

The datasets used in this work were all annotated on Swiss-Prot. However, as proven by the analysis of the Croto99 dataset, that annotation on public databases can sometimes be faulty, even when the entries are curated. In fact, a recent study by Bagheri and coworkers identified up to two million taxonomically misclassified protein entries in the NR database [134].

As much of the sequence space of enzyme families is unexplored, and annotations can be wrong even in curated databases, further research is needed to propagate annotations to novel proteins. In 2021, work by Sanchez-Pulido and Ponting showed that the prediction of tertiary structures using coevolution data can help at ”further extending the detection horizon of homology” when complemented by sequence-homology based techniques [2]. As is made evident by the novel putative connections the CSN helped reveal in this work in combination with SSNs, the analysis of enzyme families would indeed benefit from a complementary approach combining both SSNs and CSNs, which is described in this section.

As both methods have already been proven to be useful tools for making predictions on the functional linkages present in an enzyme family, using both methods side-by-side would allow for more confident assigning of annotation to novel sequences. The likely misannotation present in the Croto99 dataset is a major example of how using both SSNs and CSNs in a complementary approach can help confirm hypotheses about the function of enzymes.

Also, in this work 'optimal' thresholds were computed for the similarity networks using the available functional annotation and label propagation. However, for a dataset with minimal or no such annotation, this process is impossible. However, another benefit of using both networks as complements lies in the overall network topology similarity that SSNs and CSNs were proven to share in this work. The expected topology similarity means one could iterate multiple different thresholds for both network types and perform graph matching to identify which thresholds in the CSN are most equivalent to thresholds in the SSN. As the assumption is that annotation is lacking in this scenario, one would need to use multiple such equivalent-threshold pairs for the best result.

Undoubtedly, both SSNs and CSNs are useful tools for making predictions on the potential profile of novel sequences, but it is through the use of both methods side by side that the best understanding of an enzyme family might therefore be gained. The different meaningful linkages revealed by the approach described in this work are evidence of its viability for helping guide the selection process for more diverse enzyme panels to test in the laboratory.

3.4.3 Future work

There are four main avenues for further developing and improving the CSN method. First, this work did not delve into how optimisation of the parameters F and N would impact the quality of the resulting CSNs. It is therefore possible that whilst the quality of the CSNs at the arbitrary parameter values is undoubtedly high, parameter optimisation might further improve the quality of the linkages created by the network method. This optimisation of the parameters could also lead to a more consistent CSN range of optimal thresholds, for which rules and correlations could then be constructed.

Second, the mapping performed in section 3.2.4 is done using an MSA of the dataset from which ECPs are then extracted and filtered out using F. However, it would be interesting to see the effect of swapping to an ECP mapping that is based on pairwise alignments instead of an MSA, which is likely to be more precise as a method of associating two coevolving pairs as equivalent.

Third, while in section 3.2.5 clique-based comparisons of coevolution patterns between sequences was performed to compute the coevolution similarity score because of the findings of Lee and coworkers [144], other graph-comparison algorithms could work too. For example, it would be interesting to see how a comparison of communities produced using the Louvain method [157] would work. Relatedly, the transformation of clique similarity scores to Jaccard similarity scores is another area where a different metric could potentially work better, such as a weighted version of the Jaccard similarity score based on clique size.

Finally, the largest computational bottleneck for producing CSNs is the production of the residue-residue coevolution matric using CCMPred in section 3.2.3. While it is possible to speed up the process using powerful graphics processing unit (GPU) hardware, such resources were not accessible for this work. Such hardware would be particularly necessary to build CSNs for a much wider range of enzyme families. There is therefore potential for work on a fast mode for producing CSNs that replaces DCA-based methods for producing coevolution data with faster but less accurate methods like mutual information [158].

3.4.4 Conclusions

The features of CSNs could make them a useful tool in the functional analysis of enzyme families, especially for the goal of revealing the structural and functional diversity of a particular dataset. Specifically, CSNs can be used to correctly identify groups of similarly-functioning enzyme sequences, and to further validate currently existing functional annotation.The combination of the proven affinity CSNs have for revealing functional linkages at low sequence identity, with their notable disadvantages, means their optimal usage is as a complement to other network-based methods like SSNs. It is also important to consider a range of different cutoff thresholds in both types of networks to gain an accurate view of the enzyme family structure.

Importantly, CSNs help tackle one of the major limitations described in section 2.2.3 by not relying on raw sequence identity to ascertain functional similarity between enzymes. The use of CSNs can therefore provide another method for the enzyme family functional analysis toolbox, which can then be used to help select and refine panels of enzymes.

4

Automatic diverse subset selection from enzyme families by solving the maximum diversity problem

Contents

4.1 Introduction

The characterisation of the catalytic function of enzymes in the laboratory is currently the best means of increasing the portfolio of available biocatalysts to be used in industry. As described in section 2.2.2 of the Background, the generation of enzyme family panels of high catalytic diversity i.e. which can be applied to various chemical transformations and substrates, is a key step in optimising the enzyme characterisation process. Studies that seek to experimentally reveal novel biocatalyst space in an enzyme family generate panels using selection pipelines, whereby a subset of putative enzymes are selected from a larger set in a way that optimises the amount of catalytic diversity present [27, 32].

Such selection pipelines often necessitate manual analysis of sequence similarity data and structures like phylogenetic trees, with some using non-optimised sampling methods like random selection from clades [32]. While the analysis of enzyme families using methods like SSNs and phylogenetics has been proven to create more diverse panels, their use has known limitations. As described in section 2.2.3 of the Background, these limitations include a dependence on sequence identity, a requirement for expert knowledge about the enzyme family of interest, and a necessary time-consuming manual interpretation bottleneck.

The CSN method of performing functional analyses of enzyme families described in Chapter 3 tackles the first of these limitations, as described in section 3.4.1. However, CSNs, much like SSNs and phylogenetic trees, still require enzyme-family specific knowledge for an optimal analysis of their structures, which might not yet exist for families that are not well studied. This is especially true for the purpose of analysing enzyme families down to the level of substrate specificity, as SSNs and CSNs require specific and precise annotation to evaluate the different levels of enzyme class clustering. It is therefore not trivial to use CSNs on their own as a way of selecting candidate enzymes to be tested in the laboratory. Finally, CSNs, just like SSNs and phylogenetic trees, currently require difficult and time-consuming manual interpretation to increase the chances of choosing sequences that maximise knowledge gain in the laboratory.

Many of these limitations are further amplified the less knowledge exists about the enzymes of a dataset. Therefore, a method that can automatically sample enzymes into subsets that are catalytically diverse would be valuable. Such a method would neither rely on the existence of in-depth functional annotation, nor would it require a manual selection process to select enzymes. Moreover, these approaches can help ensure that more diverse panels of enzymes are taken into the laboratory for characterisation assays.

The method explored in this chapter took inspiration from a classic computer science optimisation problem, called the maximum diversity problem (MDP). This problem tries to solve for a set of objects the subset of K elements with the maximum diversity (or distance) given some pairwise distance metric between all chosen items [159, 160]. Practically, algorithms that try to solve the MDP take as inputs a square matrix containing the pairwise distances for all pairs of items and a number, K, for the size of the maximally diverse solution subset, and the output is simply the subset, U, of size K that is maximally diverse. It is an NP-hard combinatorial problem to solve [161], meaning that heuristic algorithms are often used to reach good solutions faster, as described in section 2.4 of the Background.

4.1.1 Algorithms for solving the MDP

Algorithms solve the MDP as a maximisation problem, where an objective function is maximised for the solution subset U and non-solution subset Z. Mathematically, solutions to the MDP are formulated as the following equation:

$$S = (x_1, x_2, ..., x_L), \text{where } x_i = \begin{cases} 1 & (x_i \in U) \\ 0 & (x_i \in Z) \end{cases} \tag{1}$$

A solution S produced by an MDP solver is simply a binary vector of size L, where L is the size of the superset to sample from, indices have a value of 1 for items included in the solution subset U, and a value of 0 if included in the non-solution subset Z. Specifically, for a subset size of K, a solution will contain K ones and $L - K$ zeros, where K is the size of the solution subset U, and $L - K$ is the size of the non-solution subset Z.

Mathematically, the objective function that is maximised is formulated as the following equation:

$$f(x) = \sum_{i=1}^{|M|} \sum_{j=i+1}^{|M|} (1 - s_{ij}), \text{where } i, j \in U \tag{2}$$

Where M is an identity matrix for some superset of items, and s_{ij} is the identity of the items i and j. The term $(1 - s_{ij})$ therefore represents the distance between two items in U rather than the identity, meaning the objective function being maximised is the sum of all distances in the subset.

There are many optimisation algorithms and paradigms that solve the MDP efficiently. For example, simulated annealing has been applied to the MDP [162], where random solutions are constantly generated as modifications of the current state at each iteration, and approved based on either performance or on a ’temperature’ based probability that goes down as the iterations progress. Evolutionary algorithms, such as memetic algorithms [163] have also been used to solve the MDP, where randomly generated solutions can be combined in ’crossover operations’, akin to chromosomal crossover in genetics. One other such heuristic paradigm that efficiently solves the MDP is tabu search, which involves features like short-term memory of recently visited solutions to evade local optima [160]. Tabu search algorithms generally solve the MDP as a set-swapping maximisation problem, where the objective function is maximised for the solution subset U through iterative swapping of items between it and the non-solution subset Z. This meta-heuristic method is known to be amongst the highest performing algorithms for solving the MDP [160].

The signature feature of tabu search algorithms for set-swapping implementations is to make recent swap moves ’illegal moves’ for a short period, or ’tabu’, unless if a move considered tabu reaches what is called the ’aspiration criterion’. This short-term memory provided by tabu moves helps move solutions away from local optima. For most tabu search implementations that solve the MDP, the aspiration criterion is simply if the objective function of a solution reached through a tabu move is higher than the current best solution.

4.1.2 Aim of this work

Many powerful algorithms that solve the MDP exist [103, 159, 160, 164, 165], but to the best of the knowledge of the author these algorithms have never been applied to the bioinformatics problem of *de novo* diverse sampling of enzyme sequences. One way of solving this problem would be to reframe it using the MDP : instead of analysing individual enzymes as potentially novel solutions, subsets of enzymes as a whole are considered as individual solutions, and their diversity is assessed based on how distant the sequences of the subset are from each other using some sequence identity metric.

It is known that proteins, enzymes included, are likely to be more similar in function and properties the more similar their sequences are. Therefore, enzyme subsets sampled by solving the MDP for a dataset would contain a high level of relative sequence diversity, and by proxy, catalytic diversity. Such a method would not require in-depth annotation about a family, and would therefore be applicable to any dataset and enzyme family independent of the level of existing knowledge. The MDP method would also provide a way of automatically selecting diverse panels of enzymes to be characterised in the laboratory, as such a diverse subset is simply the output of any algorithm that solves the MDP.

To that end, the main aim of the research described in this chapter was to explore the applicability of sampling from datasets of enzyme sequences by solving the MDP to automatically generate functionally diverse subsets. In this work, the MDP was solved using two implemented algorithms:

- A greedy heuristic algorithm called MAXMIN [164].
- A tabu search algorithm, heavily inspired by the work of Wang and co-workers [103].

Three different families with high known functional diversity and annotation were used as case studies for MDP-based sampling, and the diversity of their respective subset solutions were assessed.

4.2 Methods

4.2.1 The greedy MAXMIN algorithm

The greedy heuristic algorithm implemented is called MAXMIN [164]. The algorithm functions by iteratively making a greedy decision about which item to add to the solution set (Algorithm 2).

This algorithm is initialised by choosing the two sequences that have the highest sequence distance (and therefore lowest sequence identity) in M. Then, until the solution vector U reaches the required size K, a 'maxmin' greedy decision is made to add the next item, which first involves finding for every sequence in the non-solution subset Z the minimum distance to the sequences in U. Finally, of all such minimum distances, the sequence with the maximum distance is greedily chosen to be added to U.

```
Data: square identity matrix M, subset size K
Result: solution vector U
U = initialisation(M);
while len(S) != K do
    nextItem = chooseMaxMinItem(M, U);
    addItem(nextItem, U);
end
return U;
```

Algorithm 2: Greedy MAXMIN algorithm pseudocode

The heuristic nature of this algorithm lies in the ranking of the next item to be added using the minimum distance to the current state of the solution subset U. While it may sound counterproductive to rank based on a minimal distance when the objective is to maximise distance, it means each item is ranked based on its 'worst' contribution to the solution subset. This feature helps avoid scenarios where an item might have high distance to a single item in U but low distance to all others. This feature also guarantees that if the minimal distance for an item is still high, then it is a genuinely good contribution.

Evidently, this heuristic is also faster than attempting to find a optimal solution, because while looking for the optimal solution for the MDP is NP-hard [161], this algorithm has a polynomial running time. This algorithm is also proven to produce good solutions, especially for small values of K [166].

4.2.2 The developed tabu search algorithm

For this work, the tabu search implementation was based partly on the algorithm Wang and co-workers designed [103], with some formulaic modifications that are detailed in section 2.1.2. Pseudocode for the tabu search algorithm used to solve the MDP in this work can be found in Algorithm 3.

The algorithm starts by taking a square identity matrix M and a subset size K as inputs. From them, an initial random solution vector S is initialised, and an objective variation vector Δ is produced similarly as described by Wang and co-workers [103]. Specifically, Δ_i represents a summary value of how much a certain item in either subset U or Z varies with the items in subset U.

Next, while the exit criterion has not been reached, a neighbourhood list of two moves is produced using the same successive filter candidate list strategy as in Wang and co-workers, with a candidate list size of 10. This means that the ten indices with the highest value in Δ from both U and Z are retrieved to form candidate lists for items to be swapped. Then, all the different possible combinations of moves from U to Z using items in their respective candidate lists have their move gains computed, again as done in Wang and co-workers. Finally, the best move and the best non-tabu move are then computed using equation 2, and then added to the neighbours list.

Then, while a move has not been finalised yet, the best move from the neighbourhood is evaluated as either tabu or non-tabu. If the move is tabu, and if its objective function value is higher than the current solution's (therefore reaching the aspiration criterion), then this move swap is performed. Otherwise, the best non-tabu move is performed. After a move, changes are made to the Δ values as per Wang *et al.* Once the exit criterion is reached, then the current best solution is returned as the final output. For our work, the exit criterion is simply if 50 moves have been applied without a new best solution.

4.2.3 Modifications to Wang et al's algorithm - penalty term

One important factor in applying the MDP to protein sequence sampling is the inherent bias of datasets. Most public databases, including those such as UniProt and PFAM,

```
Data: square identity matrix M, subset size K
Result: solution vector S
S = initialisation(M, K);
Δ = initialiseDelta(S);
while exit != true do
    moved = false;
    neighbours = generateNeighbourhood(S, Δ);
    while moved != true do
        bestMove = chooseBestMove(neighbours);
        newSol = move(S, newSol);
        if bestMove.isTabu == true then
            if newSol.score > S.score then
                S = newSol;
                moved = true;
            end
        end
        else
            S = newSol;
            moved = true;
            bestMove.isTabu = true;
        end
    end
    postMoveChanges(Δ, bestMove);
end
return S;
```

Algorithm 3: Tabu search algorithm pseudocode

are likely to be made up of large clusters of very similar sequences accompanied by many smaller less-studied clusters. For example, in 2012 the 20% most common EC classes on UniProt annotated 90% of UniProt enzymes with existing annotation, with the EC class cytochrome-c oxidase (1.9.3.1) representing 12% of enzymes all by itself [167].

The crux of tabu search applications for the MDP is the naive maximisation of an objective function, which is the sum of the distances of the solution subset. With unbalanced datasets, it is therefore possible for the algorithm to consider the full inclusion of small groups of highly similar enzymes in the solution subset as optimal. This could happen due to how relatively distant the enzymes of such groups are from the larger clusters of sequences in the dataset. Obviously, for the purpose of maximising the functional diversity present in the solution subset by proxy of this objective function, this is not ideal.

Some of the functions used by Wang and co-workers [103] were therefore modified, specifically by adding a 'penalty term' P. This term is added to the objective function and the Δ calculations, and therefore all other calculations that use Δ.

$$P_{i,j} = \frac{s_{ij}}{1 - s_{ij}} \tag{3}$$

For two sequences i and j, the penalty term P_{ij} is equal to their sequence identity divided by their distance. The penalty term is such that the more similar two sequences are, the larger the penalty, and vice-versa the more distant two sequences are.

Using the penalty term, for the sequence identity matrix M, our objective function to maximise becomes:

$$f(x) = \sum_{i=1}^{M} \sum_{j=i+1}^{M} (1 - s_{ij}) - P_{ij}, \text{where } x_i, x_j \in U \tag{4}$$

The calculation of Δ for a sequence x_i becomes:

$$\Delta_i = \begin{cases} \sum_{j \in U} -(1 - s_{ij}) + P_{ij} & (x_i \in U) \\ \\ \sum_{j \in U} (1 - s_{ij}) - P_{ij} & (x_i \in Z) \end{cases} \tag{5}$$

The calculation of the move gain from swapping $x_i \in U$ and $x_j \in Z$ becomes:

$$\Delta_i + \Delta_j - (1 - s_{ij}) - P_{ij} \tag{6}$$

The post-move updates of Δ become:

$$\Delta_k = \begin{cases} -\Delta_i + (1 - s_{ij}) - P_{ij} & (k = i) \\ -\Delta_j + (1 - s_{ij}) + P_{ij} & (k = j) \\ \Delta_k + (1 - s_{ik}) - (1 - s_{jk}) - P_{ik} + P_{jk} & k \neq \{i, j\}, x \in U \\ \Delta_k - (1 - s_{ik}) + (1 - s_{jk}) + P_{ik} - P_{jk} & k \neq \{i, j\}, x \in Z \end{cases} \tag{7}$$

These modified equations will therefore guide the tabu search algorithm away from optimising for solutions that contain too many members from groups of highly similar sequences that are very distant from the rest of the superset. The penalty term was also added to the objective function the greedy MAXMIN algorithm optimises, as to compare it to the tabu search algorithm at parity. The values for the parameters used in this work can be seen in Table B.1 of the Appendix.

4.2.4 Diversity assessment - functional labels

The principal aim of this work was to produce subsets of enzyme sequences that are catalytically diverse representations of their superset. As the diversity of MDP solutions produced needs to be assessed and quantified, the way in which catalytic diversity is defined is important to clarify.

Catalytic diversity was primarily based on the level of coverage of the functional labels that is present in the solution produced by the MDP solver. Specifically, two different types of functional labels were used in this work - Enzyme Commission (EC) classes and InterPro (IP) signatures. EC classes are a curated hierarchical classification system where numeric labels representing enzyme-catalysed reactions are assigned at four progressively more specific levels of functional detail, down to the level of substrate and reaction specificity [55]. Assessing the diversity of a solution based on EC classes can therefore clearly show the many different enzymatic reactions can be catalysed by the sequences chosen.

IP signatures are conserved sequence signatures of structural or functional significance such as structural motifs and catalytic sites, that are curated by the integrated database InterPro [52]. IP signatures are not as explicit and specific to function as EC classes, which makes them worse for our purposes. However, any sequence can be annotated with the known IP signatures it contains using the tool InterProScan [54], which can help provide an overview of the diversity of a dataset when applied to all of its sequences. Therefore, IP signatures were also used in this work as a less specific but more accessible functional label for assessing the diversity of an MDP solution.

4.2.5 Diversity assessment - diversity metrics

To assess functional diversity of subsets based on the level of coverage of the functional labels described previously, two properties need to be quantified:

- Richness, which quantifies the proportion of the total classes represented by the subset, with higher richness preferred.
- Relative abundance, which quantifies the level at which classes are represented relative to each other. Higher relative abundance is also preferred, as it would imply a lack of bias and over-representation of classes.

Two metrics were therefore computed for MDP solutions that would help quantify richness and relative abundance: label coverage and the Gini-Simpson index (GSI).

The label coverage is simply the proportion of the total functional labels that exist in the superset that are covered by the sequences in the subset. Mathematically, this is defined as:

$$LC = \frac{|L_s|}{|L_t|} \tag{8}$$

Where LC is the label coverage, $|L_s|$ is the number of unique labels present in the subset, and $|L_t|$ is the total number of unique labels in the superset. The label coverage is computed for the EC classes and the IP signatures separately.

The GSI, a metric classically primarily used in ecology, is the probability that two items sampled from a set with replacement are of different classes [168]. Ranging from 0 to 1, the higher the GSI is the more likely two sampled items will be of a different class, and vice versa the lower it is. Applied to the MDP solutions produced, it becomes a measure of the relative abundance of the functional labels in the solution. Whereas the label coverage shows the level of total diversity that is present in the solution, the GSI represents how abundant each individual class is relative to the other classes in the solution. Mathematically, the GSI is defined as:

$$GSI = 1 - \sum_{i=1}^{C} p_i^2 \tag{9}$$

Where GSI is the Gini-Simpson index, C is the set of different classes in the subset, and p_i is the proportion of items in the subset that are of the ith class, meaning p_i^2 is therefore the probability that two items sampled from the subset are of the same class i. The GSI is computed for the EC classes and IP signatures separately.

An important thing to note for the GSI of IP signatures is that sequences are likely to have more than one signature, whereas sequences are unlikely to have more than one known EC class. This fact is why the term p_i is specified as being the proportion of classes of class i rather than the proportion of sequences of class i. Otherwise, all of the unique combinations of IP signatures in C would have to be considered when calculating the GSI.

4.2.6 Enzyme datasets

The enzyme datasets chosen for this work were based on four factors. First, the datasets needed to have high amounts of functional annotation that is as confidently-assigned as possible. Second, these confidently annotated datasets needed to be as large as possible, to better assess how well our method applies to large datasets. Third, the datasets needed to have as much diversity in known catalytic activities as possible. Finally, the datasets needed to be made up of bacterial sequences to avoid issues caused by large taxonomical differences within a family between eukaryotic and prokaryotic enzymes.

Bearing these limitations in mind, three ideal datasets were identified in the following manner. First, sequences were limited to Swiss-Prot, as their sequences have at least been manually curated, with annotations therefore being more confidently assigned. Then, the top 200 PFAM families were ranked based on the number of bacterial sequences on Swiss-Prot with belonging to each PFAM family. Then, sequences below 150 amino acid residues in length were removed to ignore fragments, along with sequences above 500 residues to ignore avoid multifunctional sequences. Then, any families with less than 900 sequences, a proportion of sequences annotated with full Enzyme Commission numbers below 90%, and with a GSI value for the EC classes lower than 0.7 were eliminated from consideration. Finally, of the families remaining, the top families based on the number of unique EC classes were chosen, with the top

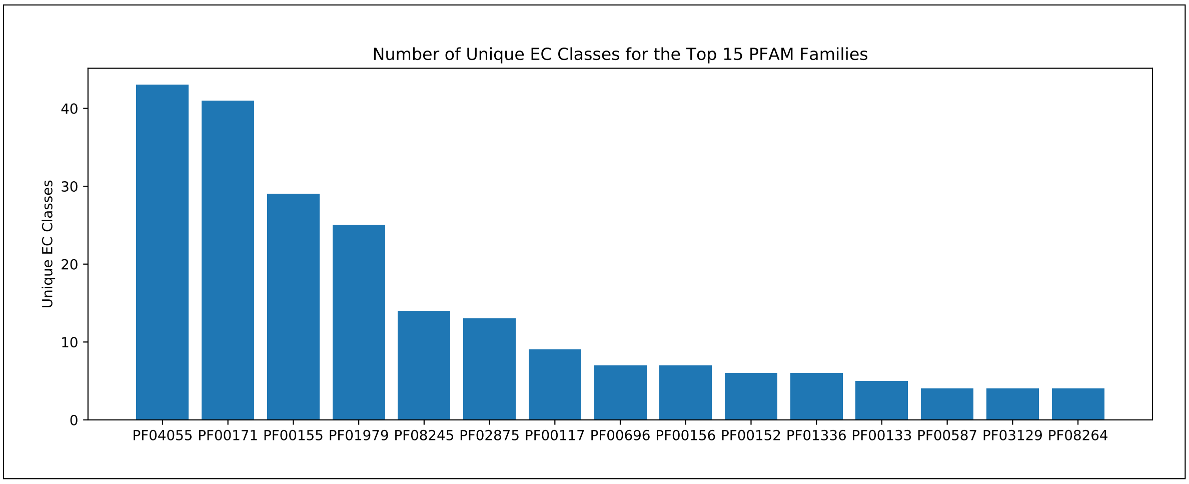


Figure 4.1: Bar plot showing the number of unique EC classes for the remaining top 15 PFAM families after filtering. Filtering was based on total number of bacterial sequences, a sequence length range between 150 and 500, a minimum number of sequences of 900, a proportion of full EC number coverage above 90%, and a minimum GSI value for the EC classes of 0.7. The top three families (PF04055, PF00171, PF00155) were chosen as the datasets for this work.

15 shown in Figure 4.1.

The top three families were picked: the radical SAM superfamily (PF04055), the aldehyde dehydrogenase family (PF00171), and the aminotransferase class I and II family (PF00155). These three datasets are referred to as SAM, ADH, and ATF, respectively for the rest of this work (Table 4.1). Importantly, the ATF dataset is the exact same dataset as Trans986 in Chapter 3.

Also, a fourth dataset was built to test the viability of the MDP method on larger datasets by retrieving 10,000 PF00155 sequences from Uniprot, which includes the 986 Swiss-Prot sequences of the ATF dataset with a further 9,014 sequences from TrEMBL. This larger dataset is referred to as ATF_TR in the rest of this chapter.

For all four of the datasets used in this work, square sequence identity matrices were built to be used as input in the MDP solver. This was done by performing all-vs-all global Needleman-Wunsch pairwise alignments, and storing all pairs of sequence identities into a square matrix.

Table 4.1: Table showing statistics about the size and functional diversity of the SAM, ATF, and ADH datasets chosen for this work.

Family	Sequence count	EC classes count	Full EC coverage	EC GSI
SAM	3105	37	97.9%	0.845
ATF	986	27	94.9%	0.689
ADH	953	35	99.0%	0.687

4.2.7 Visualising MDP solutions in the family sequence space

To visualise where MDP solutions are located in sequence space, sequence similarity networks (SSN) were generated. With sequences as nodes, they are produced by thresholding the sequence identity matrices produced for all three datasets, making edges between sequences when they are at or above that identity threshold. These networks are then visualised using Cytoscape [151].

Another method to help visualise the sequence space occupied by MDP solutions relative to their respective datasets used was the use of phylogenetic trees. The trees were generated with default parameters using FastTree [155], and then visualised using IcyTree [156].

Also, to visualise how the functional labels are covered by the solutions relative to entire datasets, signature networks were produced. For both EC classes and IP signatures, edges are made between enzymes and signatures if the sequence is annotated with said signature. These networks are then also visualised using Cytoscape [151].

4.2.8 Clustering-based comparative analysis

To further assess the value of MDP-based sampling of diverse subsets, a comparative analysis of the MDP method was performed using k-medoid clustering as a test-case [169]. The k-medoid clustering algorithm works by grouping items in a dataset based on the minimisation of the distance from the items in one group to a representative item of said group, called a medoid, which is known to help counter the effect of outliers. For the purpose of sampling K diverse sequences from a dataset using k-medoids, it is therefore as trivial as picking the medoids themselves as a solution. The sampling of k-medoids in this fashion was therefore performed in this work.

Table 4.2: Table showing the average functional and sequence diversity based on EC classes of the MDP subsets produced by running the tabu search and greedy MAXMIN algorithms on the SAM, ATF, and ADH datasets after 50 runs. The MDP was solved with a subset size K equal to 100.

Family	EC Coverage (Tabu Search)	EC Coverage (Greedy)	EC GSI (Tabu Search)	EC GSI (Greedy)	Avg SeqId (Tabu Search)	Avg SeqId (Greedy)
SAM	0.96±0.01	0.97	0.94±0.001	0.940	12.1±6.8	13.4±6.8
ATF	0.82±0.02	0.77	0.86±0.002	0.793	18.6±8.1	20.5±7.6
ADH	0.94±0.01	0.94	0.86±0.002	0.820	24.2±8.9	25.2±9.6

For this comparison, four different values of K were tested: 50, 100, 150, and 200. The performance of the MDP and k-medoid solutions were assessed based on EC label coverage and GSI.

4.3 Results

4.3.1 Analysis of the sequence diversity, and richness and relative abundance of functional classes in MDP subsets

The MDP was solved for the SAM, ADH, and ATF datasets with a solution subset size of $K = 100$, and the functional diversity of each solution subset was analysed. Because the tabu search algorithm is stochastic in nature, 50 independent runs were performed for each dataset, and the results averaged. The greedy algorithm is deterministic and therefore was only run once per dataset.

In Table 4.2, the average coverage for EC classes can be seen for all three datasets after solving the MDP with $K = 100$, and for both MDP-solving algorithms. The average pairwise sequence identity in the solution subsets can also be seen for both algorithms. In Table 4.3, the average coverage values for 1000 runs of random solutions for each dataset can also be seen.

For the tabu search algorithm results, the average EC coverage values (Table 4.2) are high for all three datasets (SAM, ADH, and ATF). For the SAM and ADH datasets, the EC coverage is above 0.90, with the lowest value amongst the three datasets being as high as 0.82 for the ATF dataset. Also, the standard deviation values are low, implying that even when runs result in different subsets of sequences being sampled, the overall diversity represented in the solutions converges to high richness.

When compared to the EC coverage richness of randomly chosen solutions, which can be seen in Table 4.3, the tabu search MDP solutions outperform their respective random solutions by +0.65, +0.41, and +0.6 for SAM, ATF, and ADH, respectively. There also is not as high a convergence in overall diversity for the random solutions, as shown by the higher overall standard deviations. This result indicates that using the tabu search algorithm for solving the MDP results in subsets of higher EC richness compared to random sampling, which is a simple initial test of performance.

The greedy algorithm solutions would be expected to have lower richness than the tabu search [160, 166]. However, for the three datasets, the two algorithms perform evenly, except for the ATF dataset, where the tabu search algorithm has +0.05 EC label coverage. This higher than expected parity is likely partly explained by the addition of the penalty term to the greedy MAXMIN algorithm. Expectedly, the greedy solution subsets also have significantly higher richness than those produced through random sampling, as can be seen in Table 4.3, with higher EC coverage by +0.66, +0.36, and +0.6 for the SAM, ATF, and ADH datasets, respectively.

These coverage results show that solving the MDP to sample sequences creates subsets that cover a majority of the known functional labels in these datasets, and that therefore have high richness in terms of functional diversity. These results are also consistent across the datasets, as the MDP solutions for all three datasets contain high richness. Also, when comparing the tabu search and greedy algorithms, it is shown that the tabu search MDP solver samples subsets of sequences with higher richness than the greedy algorithm for one of the three datasets; they are otherwise similar in performance.

Table 4.3: Table showing the average functional diversity based on EC classes and InterPro signatures of the subsets produced by randomly sampling subsets of size equal to 100 for the SAM, ATF, and ADH datasets, after 1000 runs. The functional diversity present in randomly sampled subsets is significantly worse than those created by both MDP algorithms, as can be seen in Tables 4.2 and 4.5.

Family	EC Coverage	EC GSI	IP Coverage	IP GSI
SAM	0.31±0.04	0.83±0.011	0.43±0.053	0.945±0.01
ATF	0.41±0.071	0.68±0.032	0.59±0.05	0.911±0.001
ADH	0.34±0.05	0.68±0.045	0.61±0.059	0.918±0.001

In Table 4.2, the average GSI for both EC classes can be seen for all three datasets after solving the MDP with $K = 100$, again for both algorithms. Just as with the EC coverage, the average EC GSI values are high for all three datasets, with average GSI as high as 0.94 and as low as 0.86 for the tabu search solutions, and as high as 0.94 and as low as 0.793 for the greedy heuristic solutions. More specifically, the tabu search solutions have higher GSI values for both the ATF and ADH datasets, by margins of +0.067 and +0.04, respectively. Therefore, for relative the tabu search algorithm is seen to marginally outperform the greedy algorithm for two out of three datasets.

The GSI is a measure of relative abundance of classes, and therefore comparing the GSI of the solution subsets to the original datasets (Table 4.1) can help assess the amount of diversity in the solution subsets. It is shown that for the tabu search MDP-sampled subsets, the average EC GSI values are significantly higher than those of the original datasets, with performances better by +0.095, +0.171, and 0.173 for SAM, ATF, and ADH, respectively. The solutions of the lower-performing greedy algorithm also have higher relative abundance than the original datasets, with higher GSI values by +0.095, +0.104, and +0.133.

Finally, in Table 4.2 the average pairwise sequence identity of sequences in the solution subsets can be seen. This metric gives an idea of how each algorithm has maximised the objective function itself i.e. the amount of sequence distance between items included in the solution subset. For all three datasets, the tabu search algorithm outperforms the greedy algorithm, by margins of +1.3, +1.9, and +1.0 for the SAM, ATF, and ADH datasets, respectively. This result implies that while the two algorithms have similar performances in choosing datasets of high richness and relative abundance of EC classes, the tabu search method does produce subsets with higher raw sequence distance between them. Owing to the higher performance across this metric of the tabu search algorithm, along with marginally better relative abundance of classes, this method is used as the basis of the rest of the results of this section.

The highest performing run for every dataset was retrieved, and signature networks with the selected sequences highlighted can be seen for each dataset in Figure 4.2. These act as visual representations of this significant improvement in GSI for the subsets. For all three datasets, the distribution of nodes representing each EC class

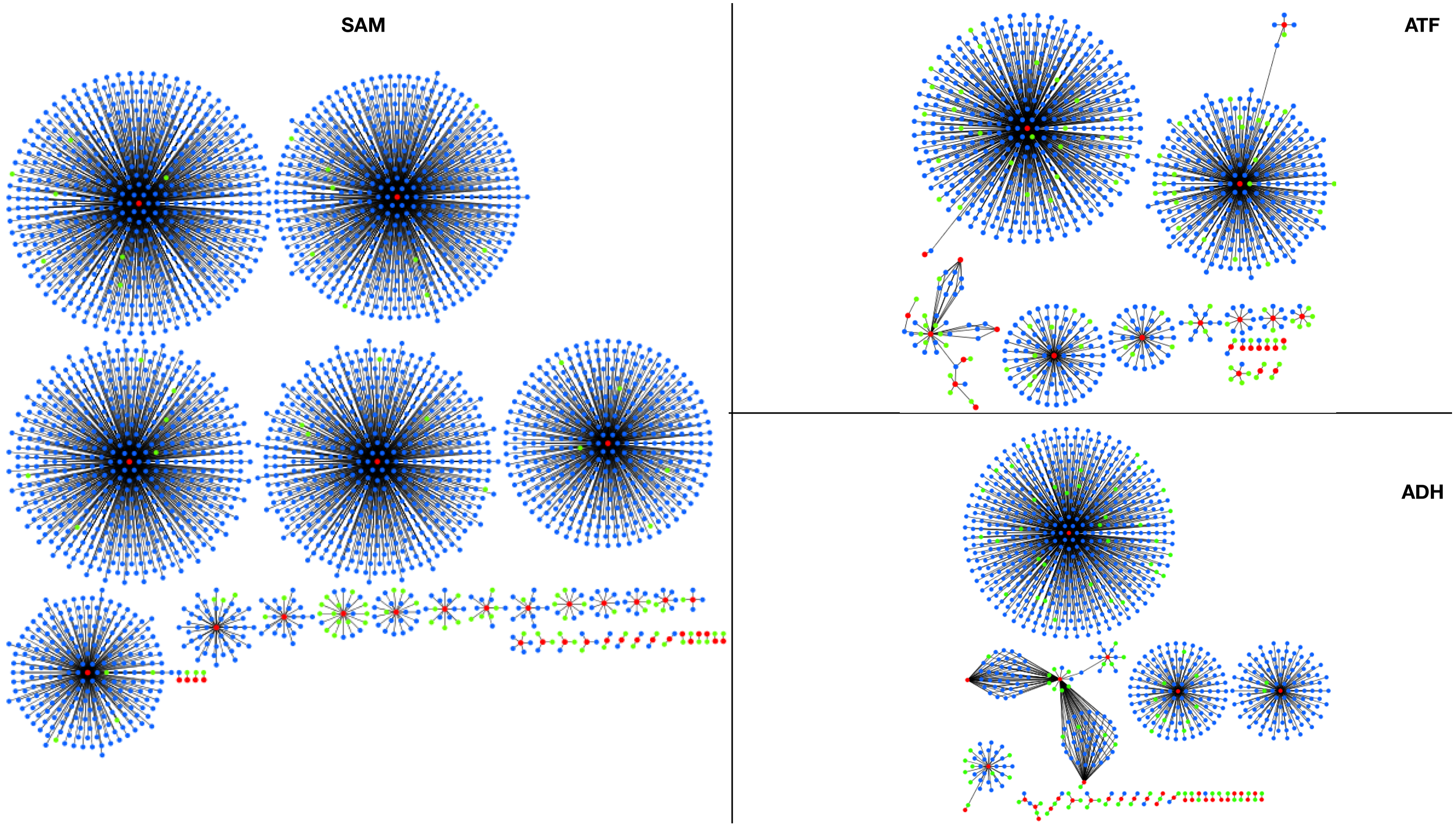


Figure 4.2: EC class signature networks for the SAM, ATF, and ADH datasets with the tabu search MDP subsets highlighted. Signatures networks are made up of enzyme (blue nodes) and EC nodes (red), and edges are made between these two types of nodes if some enzyme has some EC class. Red nodes are selected by the tabu search MDP algorithm. These networks help visualise the spread of the functional diversity present in the MDP subsets compared to the total diversity of the dataset.

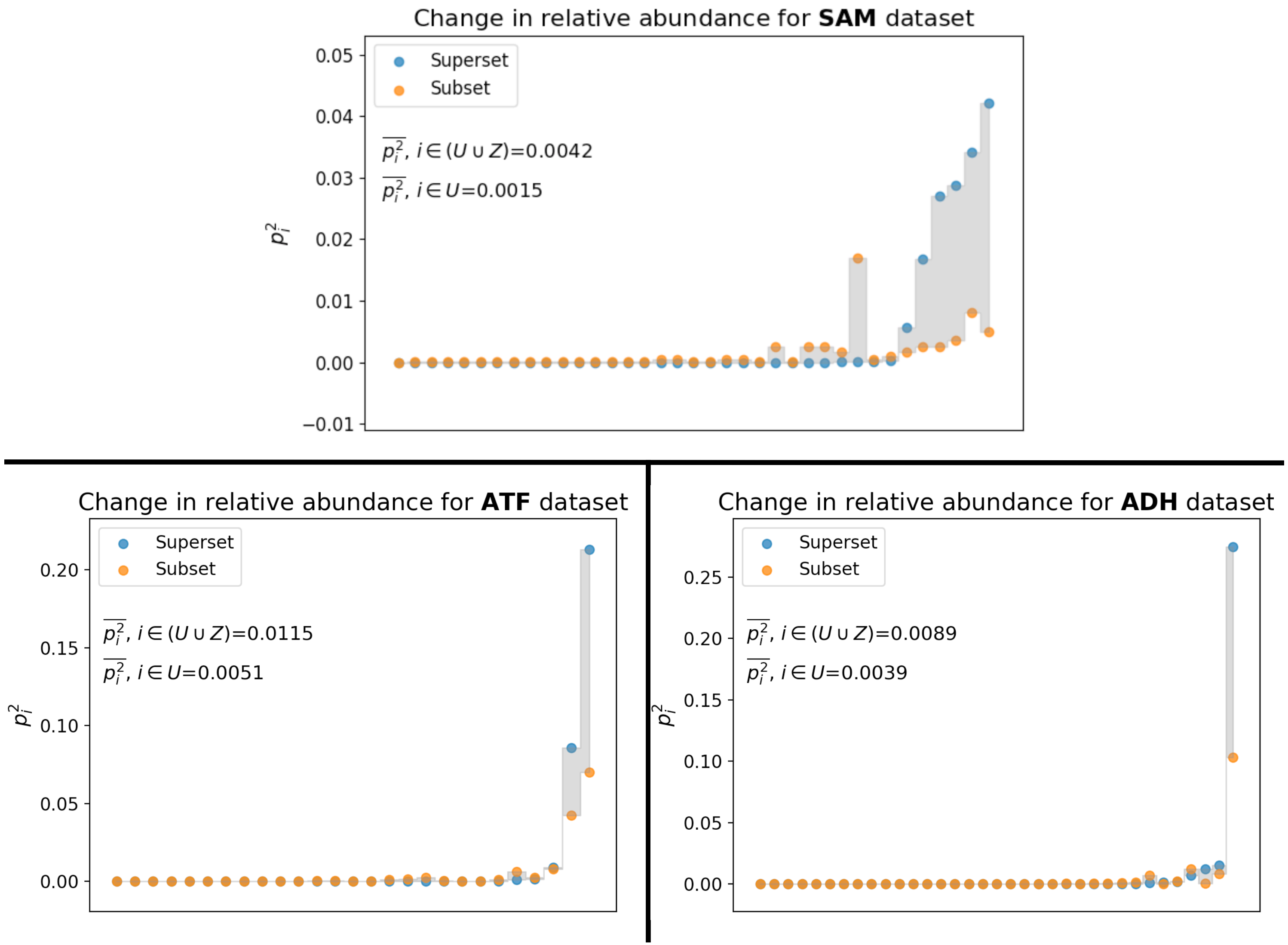


Figure 4.3: Scatterplots showing the change in relative abundance for EC classes after tabu search based MDP-sampling of the SAM, ATF, and ADH datasets. Blue and orange data points represent EC classes, and their class abundance (p_i^2 in equation 9) in the superset and the MDP subset is plotted. The average EC class abundance can also be seen for the superset ($\overline{p_i^2}$, $i \in U$) and the subset($\overline{p_i^2}$, $i \in (U \cup Z)$). A pattern of EC classes that are highly represented in the superset drastically being less abundant in the subset can be seen, in favour of further representation for less common classes. This is also shown by the average class abundance in the subset being smaller than in the superset for all three datasets.

is relatively even - larger connected components do not overwhelm the membership of the solution subset, and the vast majority of smaller connected components were represented even when they only have one node per EC class. These networks therefore showed, just like the GSI values, that the tabu search MDP solutions have a high relative abundance of EC classes compared to the total diversity of the datasets.

In Figure 4.3, scatterplots show how each individual EC class' abundance (p_i^2 in equation 9) changes from the superset to the solution subset for all three datasets. It can be seen on these plots that for every dataset, if a class was originally highly represented in the superset, its level of membership in the solution subset gets significantly reduced, without being absent. With the abundance of such classes going down, less common classes were represented in higher numbers in the solution subset. This change in distribution of relative abundance is also seen in the change in the mean value of p_i^2 from the superset to the subset - for all three datasets, $\overline{p_i^2}$ is significantly smaller for the solution subset (Figure 4.3). This result therefore also shows the higher and more equitable relative abundance of EC classes in the MDP solutions.

To summarise, the MDP algorithm, and in particular the tabu search algorithm developed, was shown to have high relative abundance of the known catalytic classes on top of the high richness observed previously. Specifically, MDP solutions showed high GSI values of solution subsets, especially when compared to the supersets, and showed changes in individual GSI contributions that increase overall abundance of rarer classes at the expense of over-represented classes.

4.3.2 Analysis of similarity relationships between MDP-sampled sequences

In Fig 4.4, SSNs at 40% identity threshold can be seen for the three datasets studied, along with the solution subgraphs made up of the MDP solution subset nodes (in green) for the highest performing run of each dataset. This figure also shows the network density of each network and its respective solution subgraph.

It can be seen that for all three SSNs, selected nodes are spread out across the topology of the network, with almost every connected component having selected nodes at this threshold. As such components can be interpreted as areas of likely protein homology,

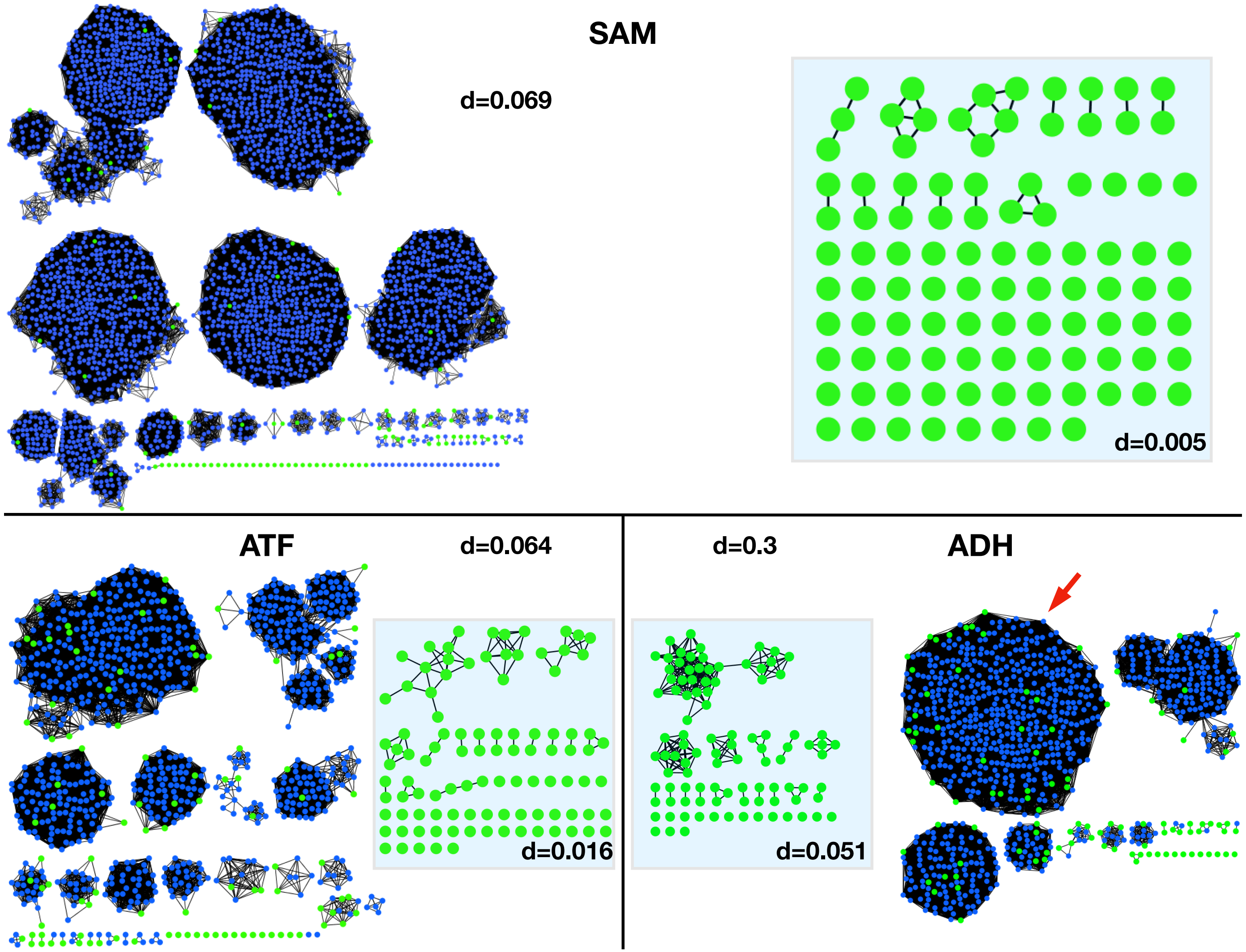


Figure 4.4: SSNs at a 40% identity threshold for the SAM, ATF, and ADH datasets. Green nodes represent the enzymes sampled by the tabu search MDP algorithm, the subgraph of which can also be seen. The network density for the SSNs and the MDP solution subgraphs can also be seen. In this case, the SSNs help show the evolutionary spread of the MDP subsets relative to the supersets. The spread of selected nodes across the different components of the SSN is high, with almost every component being represented by at least one node. Also, the density of the solution subgraphs is lower than their respective graphs for all three datasets, meaning that on average sequences have less in common in the subgraphs than in the original graphs. However, the value of this subgraph density is dependent on the original density of the graph, as evidenced by the far larger density of the ADH SSN, which is mostly due to the large component pointed at by a red arrow that contains more than half of the ADH nodes.

a diverse solution subset will contain sequences from many of these individual components. Such a spread in the network topology would therefore represent a minimisation of homologous relationships between sampled sequences, and therefore maximising the functional diversity between them, which is indeed seen here.

Also, while differing thresholds can very quickly connect edgeless nodes to the rest of the network, almost all edgeless nodes present at 40% identity are present in their respective solution subsets: the SAM subset contains 29 of its 49 edgeless nodes (59%), the ATF subset contains 15/17 of the edgeless nodes (88%), and the ADH subset contains all 11 edgeless nodes (100%). This high representation of edgeless nodes in solution subsets is another indication that the solutions produced maximise sequence (and therefore functional) diversity, as edgeless nodes are likely to be some of the members most distant from the rest of a dataset for that threshold. These sequences would therefore be more likely to contribute functional diversity when included in solutions.

It can also be shown that the solution subgraphs have lower network densities than their original datasets, meaning that on average sequences have less in common in the subgraphs than in the original graphs. However, the density of the overall solution subgraph seems to depend on the density of the original network, and therefore of the sequence bias of the original dataset. Indeed, 56.9% of ADH sequences belong to a single highly dense component at 40% threshold (indicated by a red arrow) - implying high sequence bias - has the highest density of all three solution subgraphs, with only 16% of its nodes being edgeless. As for the SAM dataset - which has a far lower density, higher EC GSI of 0.845, more unique EC classes, and more than triple the amount of sequences - its subgraph has far lower network density, with 67% of its nodes being edgeless. Nonetheless, relative to this sequence bias, the ADH solution subgraph still contains all of the topological indicators of a diverse solution subset - with a far lower density than the original network, with all of its edgeless nodes being selected, and all of its connected components being well-represented.

In Figures 4.5, 4.6, and 4.7, the phylogenetic trees can be seen for each dataset. Branches coloured in red represent the sampled sequences of the best performing runs and their respective ancestor nodes. It can be seen that on all three trees there is

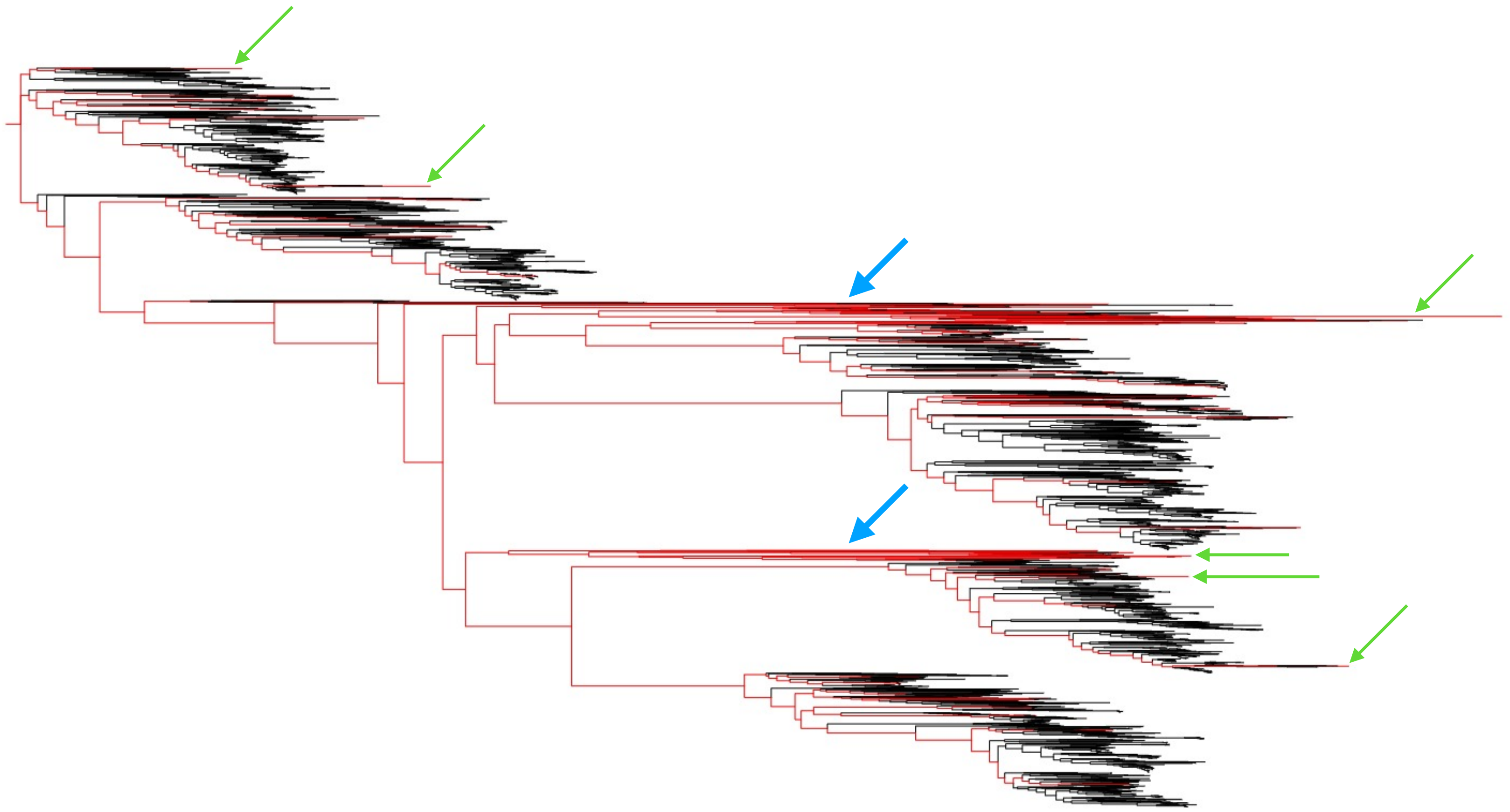


Figure 4.5: Phylogenetic tree for the SAM dataset, with branches coloured in red representing the enzymes selected by the tabu search MDP algorithm along with the respective ancestors nodes, showing a high amount of evolutionary spread. Green arrows point out sequences that were chosen by the MDP algorithm that are also evolutionarily the furthest away leaf in their respective clades, showing that the MDP algorithm is good at maximising distance even within clades. Blue arrows point out some areas in the tree in which clades are overrepresented relative to other clades, showing that the selection could still perform better.

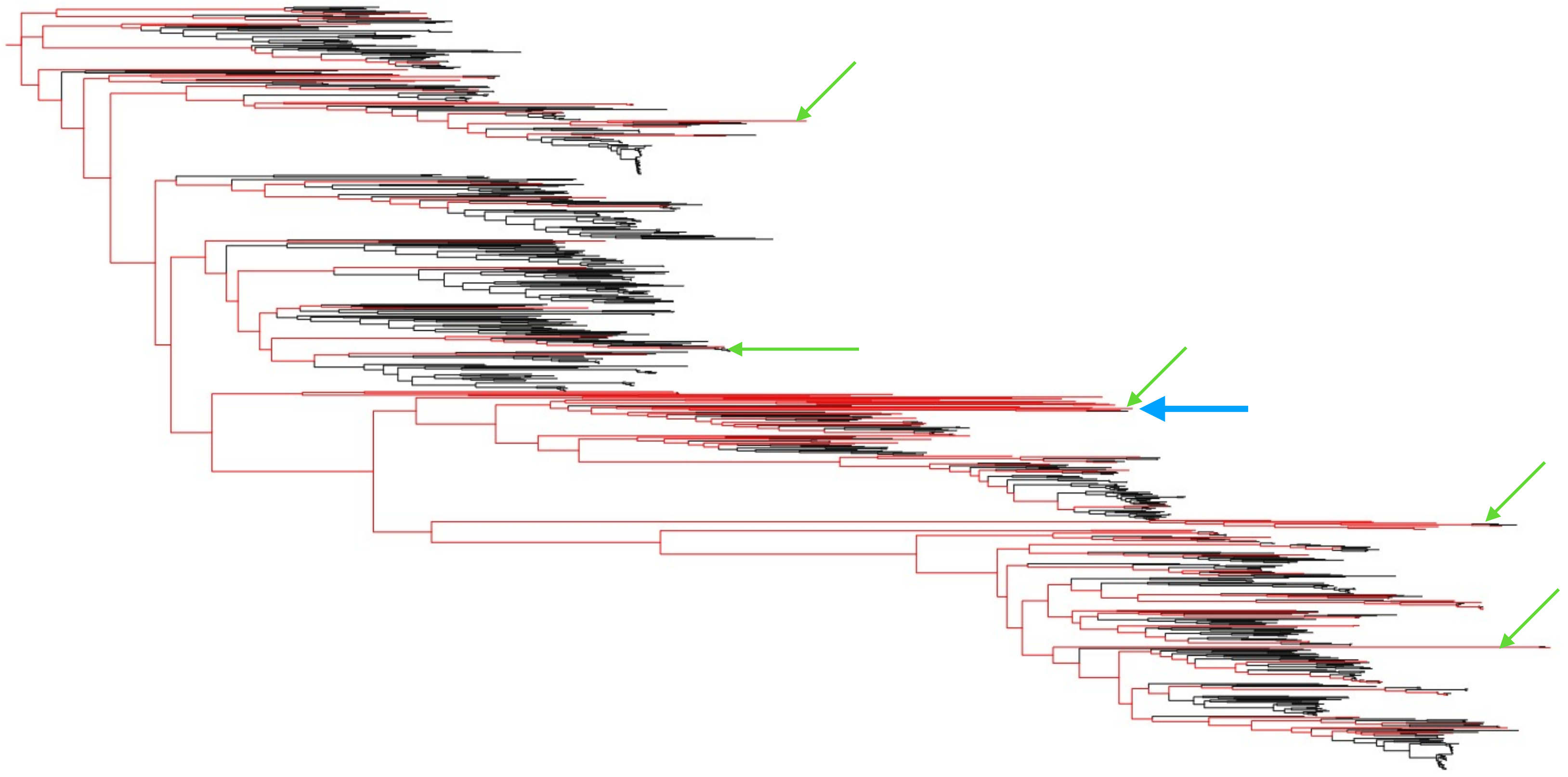


Figure 4.6: Phylogenetic tree for the ATF dataset, with branches coloured in red representing the enzymes selected by the tabu search MDP algorithm along with the respective ancestors nodes, showing a high amount of evolutionary spread. Green arrows point out sequences that were chosen by the MDP algorithm that are also evolutionarily the furthest away leaf in their respective clades, showing that the MDP algorithm is good at maximising distance even within clades. Blue arrows point out some areas in the tree in which clades are overrepresented relative to other clades, showing that the selection could still perform better.

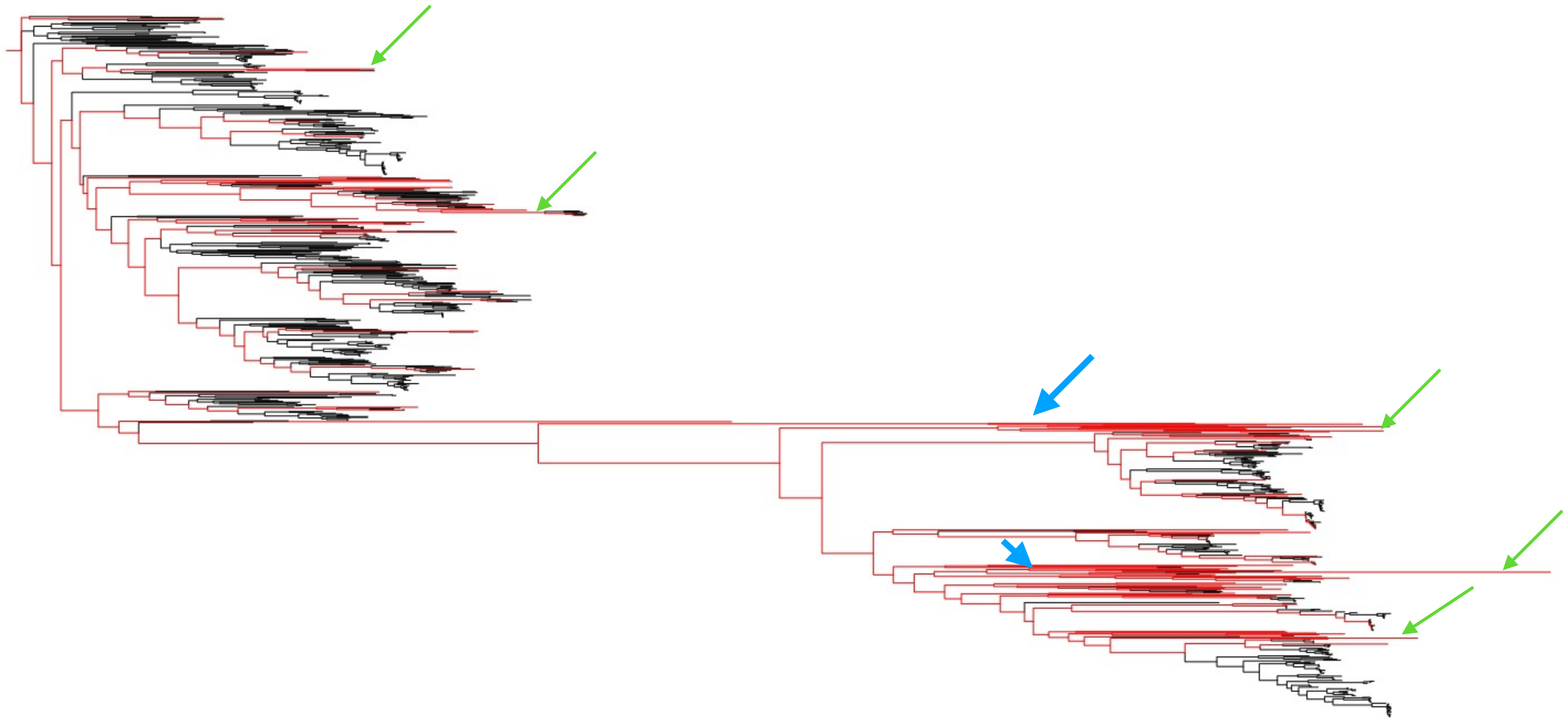


Figure 4.7: Phylogenetic tree for the ADH dataset, with branches coloured in red representing the enzymes selected by the tabu search MDP algorithm along with the respective ancestors nodes, showing a high amount of evolutionary spread. Green arrows point out sequences that were chosen by the MDP algorithm that are also evolutionarily the furthest away leaf in their respective clades, showing that the MDP algorithm is good at maximising distance even within clades. Blue arrows point out some areas in the tree in which clades are overrepresented relative to other clades, showing that the selection could still perform better.

a high amount of evolutionary spread, with all major clades represented as shown by the almost complete highlighting of ancestor nodes across all three trees. Another observation is that in every clade, leaves furthest away in terms of evolutionary change from the rest tend to be picked by the MDP algorithm, as shown by the green arrows. Such an observation implies the MDP algorithm does prefer sequences at the edges of sequence space, even at the individual clade level.

However, evolutionary bias in terms of the sampled sequences is also clearly present, with some clades being overrepresented, which are indicated by blue arrows. The sampled sequences could therefore optimally be more spread out, as such over-representation of clades is unlikely to be optimal. One hypothesis to explain this over-representation of certain clades is that the penalty term introduced in section 4.2.3 does not perform optimally. Specifically, the trade-off of fully selecting highly similar clusters that are highly dissimilar to the rest of the subset still produces a mathematically better solution in terms of the sum of distances.

4.3.3 Comparison of MDP-based sampling with k-medoids

It is important to assess how the MDP-based method of sampling functionally diverse subsets competes with other known methods. k-medoids is a clustering technique that can be used to sample from a dataset by solving it with k being equal to the subset size K and then extracting the medoids themselves as the members of the subset.

For a thorough comparison of the two competing methods, four different values for K were studied: 50, 100, 150, and 200. The tabu search MDP was run 50 times for each value of K, with the results averaged up. The result of these runs for both methods can be seen in Table 4.4.

For the EC class coverage, there are 12 comparisons made here - three datasets for each value of K. Across these, the MDP method has higher coverage of the EC classes in 10 of them, with a margin ranging from 0.029 to +0.37. The MDP has higher richness in terms of EC classes across all three datasets for K=50 and K=100, and for two of the datasets for K=150 and K=200. The other two comparisons where the MDP does not outperform the k-medoids is K=150 and K=200 for the ATF dataset, where the

Table 4.4: Table showing the functional diversity of the tabu search MDP-based solutions in comparison to the k-medoids sampling for values of K equal to 50, 100, 150, 200. For the EC coverage, the MDP method outperforms the k-medoids method in 10 of the 12 different comparisons. For the GSI, the MDP method outperforms the k-medoids method in 9 of the 12 comparisons.

	MDP (Tabu Search)			**k-medoids**		
	Family	**EC Coverage**	**GSI**	**Family**	**EC Coverage**	**GSI**
K=50	**SAM**	0.778±0.019	0.948±0.001	**SAM**	0.54	0.89
	ATF	0.70±0.02	0.88±0.005	**ATF**	0.333	0.627
	ADH	0.62±0.02	0.869±0.003	**ADH**	0.342	0.738
K=100	**SAM**	0.96±0.01	0.94±0.001	**SAM**	0.73	0.89
	ATF	0.82±0.02	0.86±0.002	**ATF**	0.52	0.66
	ADH	0.94±0.01	0.86±0.002	**ADH**	0.63	0.86
K=150	**SAM**	0.979±0.0	0.941±0.0007	**SAM**	0.783	0.904
	ATF	0.87±0.01	0.83±0.002	**ATF**	0.888	0.779
	ADH	0.988±0.01	0.846±0.001	**ADH**	0.828	0.859
K=200	**SAM**	0.982±0.012	0.941±0.0007	**SAM**	0.89	0.9
	ATF	0.962±0.005	0.82±0.001	**ATF**	0.962	0.806
	ADH	1±0.0	0.82±0.01	**ADH**	0.971	0.862

two methods are seemingly even. These results therefore seem to indicate that tabu search-based MDP solutions have significantly higher richness than the k-medoid ones.

As for relative abundance, there are once again 12 comparisons. The MDP solutions are once again better in 9 of the 12 comparisons, with margins of the GSI ranging from 0.014 to 0.26. The MDP algorithm yet again unanimously outperforms the k-medoid for K=50, and for two of the datasets for the other three values of K. For two of the last three comparisons, the GSI is even between the two methods, while for K=200 for the ADH dataset the k-medoid has a higher GSI by 0.042. However, the label coverage for the MDP solution for this scenario is at 100%, a state of full coverage it likely achieves at a subset size smaller than 200. Therefore, the MDP algorithm substantially outperforms the k-medoids in both richness and relative abundance for subset selection. The reason is likely because any MDP algorithm more directly maximises the distance between members of a subset, compared to k-medoids which is at its core a clustering algorithm.

Table 4.5: Table showing the average functional diversity based on InterPro signatures of the MDP subsets produced by running the tabu search and greedy MAXMIN algorithms on the SAM, ATF, and ADH datasets after 50 runs. The MDP was solved with a subset size K equal to 100. Unlike with EC classes, the tabu search algorithm is not shown to be unanimously superior to the greedy algorithm for every dataset (Table 4.2), though the average richness and relative abundance are still high for both.

Family	IP Coverage (Tabu Search)	IP Coverage (Greedy)	IP GSI (Tabu Search)	IP GSI (Greedy)
SAM	0.89±0.009	0.94	0.95±0.0003	0.955
ATF	0.89±0.01	0.92	0.91±0.0003	0.91
ADH	0.91±0.008	0.98	0.91±0.0002	0.91

4.3.4 InterPro signatures as de novo functional labels for subset analysis

As confident EC labels are scarce, since they require experimental evidence, the use of IP signatures as a stand-in for unannotated data was assessed. As described in section 2.3.1 of the Background, IP signatures can be identified *de novo* for sequences and can therefore help give an idea of the amount of diversity in a dataset and resulting subsets when knowledge about individual sequences is scarce.

Ideally, IP signatures are therefore a direct replacement to EC labels for the assessment of functional diversity in a dataset. For these signatures to qualify as such, the three patterns in EC-assessed MDP solutions revealed in section 4.3.1 must therefore be conserved:

1. The solutions have high richness in IP signature label coverage
2. The solutions have high relative abundance in IP signature GSI
3. The solutions produced by the tabu search algorithm are as good or marginally better than the greedy heuristic solutions across all three datasets

In Table 4.5, the average coverage of IP signatures can be seen for all three datasets for both MDP algorithms after solving it with $K = 100$. Once again, 50 runs of the tabu search algorithm were averaged due to its stochastic nature.

Much like with EC classes, the label coverage when using IP signatures is high across all three datasets and for both algorithms. For the tabu search subsets, the coverage ranges from 0.89 to 0.91, while for the greedy algorithm it ranges from 0.92 to 0.98.

There is a higher state of solution convergence than with EC labels, as shown by the even lower standard deviations of the coverage. When compared to the IP signature coverage of randomly selected subsets (Table 4.3), the MDP solutions again have higher richness. These results therefore clearly imply a high richness of IP signatures achieved in the MDP-produced subsets.

Also, for both algorithms and for all datasets, the IP GSI is high, ranging from 0.91 to 0.95 for both the tabu search and greedy solutions. It is worth noting that, compared to the GSI of randomly selected subsets (Table 4.3), those values are the same. This observation is likely due to the inherently high number of signatures in each dataset, combined with the fact that sequences will have more than one IP signature each, unlike with EC classes. This high number of signatures makes it therefore already unlikely that two randomly selected classes from a subset would be of the same class, resulting in a high IP GSI for the randomly selected subsets. One implication of this result is that the GSI might not be well-suited to assess relative abundance of classes when the number of different classes is high. Nonetheless, the relative abundance of the signatures is clearly high for the MDP solutions, even with this limitation.

The final pattern that needs to be conserved for IP signatures to make a reliable replacement for EC classes for diversity assessment, is for the tabu search algorithm to have a similar performance as the greedy algorithm in terms of the richness and relative abundance of their respective solutions. However, this pattern is not conserved. Indeed, while the coverage values are high as a whole for both algorithms, the greedy solution has higher IP signature coverage than the tabu search solution for two of the three datasets.

In Table 4.6 further details of the coverage of IP signatures in the different solutions can be seen for the highest performing run of each dataset: the number of IP signatures shared between both the tabu search and greedy solutions (intersection), those unique to both, and those not covered by either of them. For all three datasets, the majority of the signatures were covered by both the tabu search and greedy solutions, with the intersection coverage ranging from 85.6% to 90.9%. As the results in Table 4.5 showed, the greedy search solutions have slightly higher coverage for the SAM and ADH datasets, with up to 8.8% of the SAM IP signatures being uniquely covered by

Table 4.6: Table showing the coverage of IP signatures for the MDP solutions produced by the tabu search and greedy algorithms for all three datasets, including overlap and signatures uniquely covered by each method, for the SAM, ATF, and ADH datasets. While most IP signatures are covered by both methods, the greedy solutions have two more uniquely covered signatures than the tabu search solutions across the three datasets for a total of 10 versus 8. Only a minority of signatures are not covered.

	SAM	**ATF**	**ADH**
Intersection of Signatures	107 (85.6%)	49 (89.0%)	50 (90.9%)
Unique Tabu Search Signatures	3 (2.4%)	0 (0%)	1 (1.8%)
Unique Greedy Signatures	11 (8.8%)	2 (3.6%)	4 (7.2%)
Signatures Not Covered	4 (3.2%)	3 (5.4%)	0 (0%)
Total	125	55	55

the greedy solutions. However, both the tabu search and the greedy solutions have signatures unique to their solutions: in total across all three datasets, the tabu search solutions contain 4 unique signatures, while the greedy solutions contain 17. Finally, only a minority of signatures are covered by neither method, with the uncovered signatures ranging from 0% to 5.4%, showing that both methods produce solutions with a high level of richness for IP signatures.

To examine the potential reasons for a higher performance parity between the tabu search and greedy algorithm when using IP signatures, signature networks were created for their respective solutions, which can be seen in Figure 4.8. Orange nodes are signatures that are present in both solutions (or in the intersection), green nodes are unique to the tabu search solutions, pink nodes are uniquely covered by the greedy algorithm solutions, blue nodes were not in either solution, and finally black nodes are enzymes. As shown in Table 4.6, the vast majority of signature nodes are orange and therefore covered by both methods. The vast majority of signatures in the intersection have very high degree, and the most high degree signatures tend to be in the intersection.

For example, in Table 4.7 further information can be seen on the signatures that have the highest degree in the three networks shown in Figure 4.8, including for both the intersection nodes and the nodes unique to MDP solutions. The top intersection signature nodes have the highest degree possible for their respective networks i.e. every sequence in a dataset has these respective signatures as annotation. The SAM dataset has one signature at the maximum degree, while ADH and ATF both have three

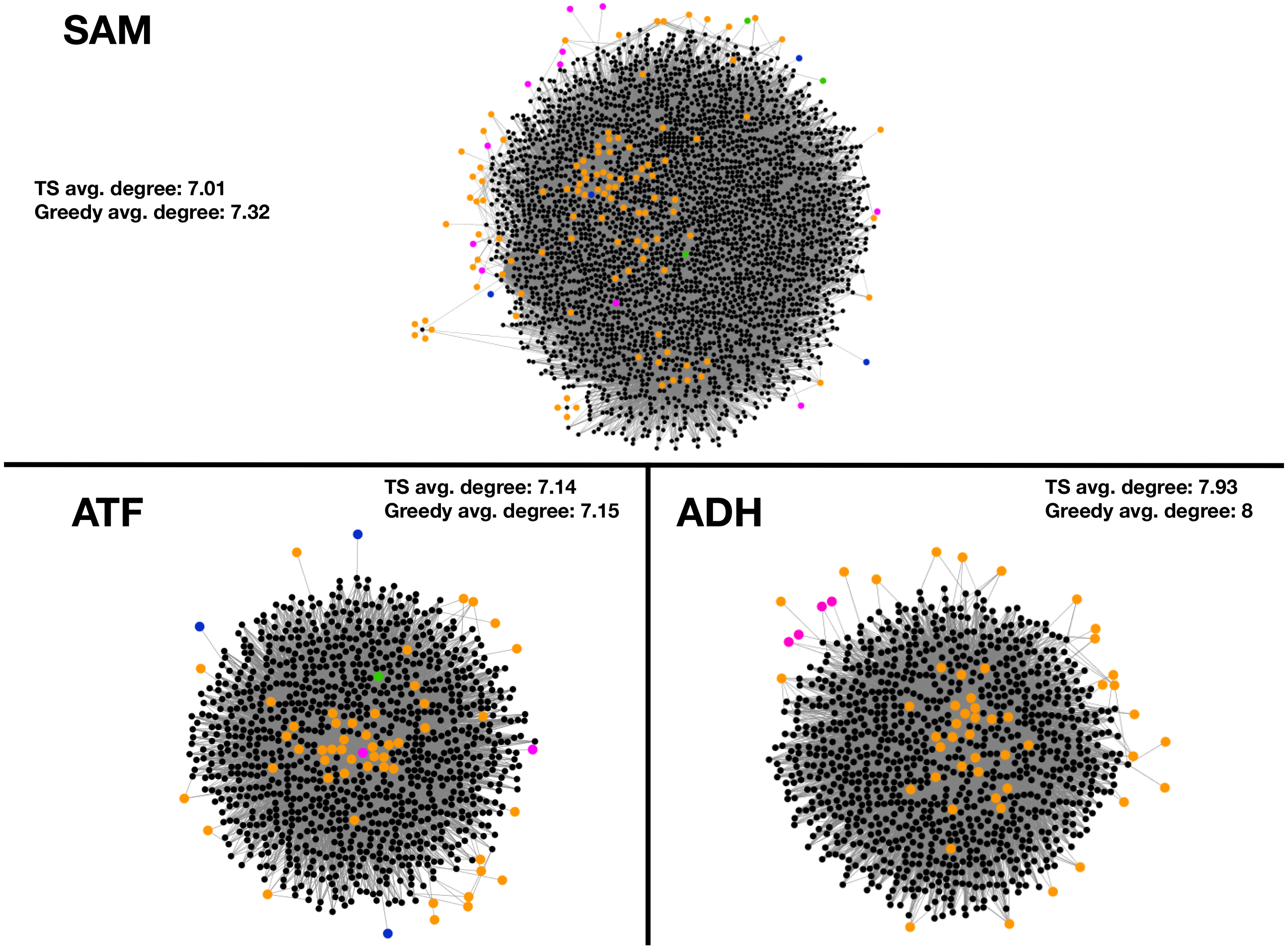


Figure 4.8: InterPro signature networks for the SAM, ATF, and ADH datasets showing a comparison of the IP signatures covered by the MDP subsets chosen by the tabu search and greedy algorithms. These signatures networks are made up of enzyme and IP signature nodes, and edges are made between these two types of nodes if some enzyme has some IP signature. Orange nodes are signatures that are present in both solutions, green nodes are unique to the tabu search solutions, pink nodes are uniquely covered by the greedy algorithm solutions, blue nodes were not in either solution, and black nodes are enzymes. Most signatures in the intersection have high degree, and almost all high degree nodes are part of the intersection. Signatures covered uniquely by the tabu search subsets have on average lower degree than the greedy subsets across all three datasets.

signatures at the maximum degree. From the description of these signatures, and the fact that every enzyme of each respective dataset bears these signatures, they simply represent each of the three enzyme families used in this work. In fact, the signatures IPR007197, IPR004829, and IPR015590 are the InterPro IDs given to the PFAM ID of each respective dataset, which were used to build the datasets in the first place.

Table 4.7: Table showing detailed information about the highest and lowest degree InterPro signature nodes for the intersection and for the signatures unique to the MDP solutions for the SAM, ATF, and ADH datasets.

	Family	Signature	Type	Description	Degree
Intersection Signatures	**SAM**	IPR007197	Domain	Radical SAM	3105
	ATF	IPR004839	Domain	Aminotransferase, class I/classII	986
	ATF	IPR015421	Homologous Superfamily	Pyridoxal phosphate-dependent transferase, major domain	986
	ATF	IPR015424	Homologous Superfamily	Pyridoxal phosphate-dependent transferase	986
	ADH	IPR015590	Domain	Aldehyde dehydrogenase domain	953
	ADH	IPR016161	Homologous Superfamily	Aldehyde/histidinol dehydrogenase	953
	ADH	IPR016162	Homologous Superfamily	Aldehyde dehydrogenase, N-terminal	953
Unique Signatures	**SAM**	PTHR10949:SF0	PANTHER Family	Lipoyl Synthase, Mitochondrial	42
	ATF	IPR024892	Family	Putative phenylalanine aminotransferase	31
	ADH	PTHR43521	PANTHER Family	Alpha-aminoadipic semialdehyde dehydrogenase	2

As for the signatures unique to MDP solutions, the result is different. First of all, the degrees are far lower, at node degrees of 42, 31, and 2 for the SAM, ATF, and ADH datasets, respectively. Second of all, while the intersection signatures of a dataset are general to the sequences of that dataset, these signatures are specific in description - the SAM signature is an evolutionary subfamily specific to mitochondrial machinery, the ATF signature is a specific type of aminotransferase, while the ADH signature is a specific type of dehydrogenase with a very small degree of 2.

This difference between intersection signatures and those unique to MDP solutions, combined with the fact that both algorithms produce solutions with uniquely covered signatures, points to an explanation involving the lack of completeness of IP signatures. As the InterPro database is continuously integrating more signatures based on the proteins that get researched, and that these can be biased, it means that certain enzymes will have a more complete set of IP signatures than others.

Also, it is known from the results in section 4.3.2 that the tabu search algorithm tends to select sequences further at the edges of sequence similarity compared to the greedy algorithm. With respect to IP signatures, it is therefore likely that sequences with less annotation are picked by the tabu search solver at a higher frequency than the greedy algorithm. Indeed, the average node degree of sequences for all three datasets can be seen in Figure 4.8. As suspected, the average degree of sequences selected by the tabu search algorithm is lower than for the greedy algorithm across all three datasets, especially for SAM and ADH. This result therefore implies that sequences in tabu search solutions have less curated IP signatures on average, likely due to a higher proportion of sequences at the edge of the respective family.

In conclusion, it is therefore unlikely that InterPro signatures can be used as de novo functional labels on par with EC classes due to incompleteness and annotation bias. However, this is likely to depend on how well studied sequences of interest are, with the assumption that the more studied an enzyme family is, the more spread out recognisable sequence signatures will be. Also, these results with IP signatures are further evidence of MDP-based solutions sampling diverse subsets in terms of both richness and relative abundance of the annotations given. Finally, while IP signatures are less specific than EC classes overall, they are far easier to produce for novel sequences or sequences that are not as well studied, and therefore still qualify as a good annotation for the assessment of MDP-sampled subsets when EC classes are impossible to retrieve.

4.3.5 Evaluation of MDP-based subsets from larger datasets

The ideal use case for diverse sampling of enzymes by solving the MDP is large unannotated datasets from which smaller diverse subsets can be selected for further exploration and analysis. Therefore, to evaluate the utility of MDP-based subset selection, the ATF_TR dataset, which contains 10,000 enzymes, was compiled and analysed with respect to MDP solutions generated by both the greedy and tabu search algorithm. As discussed before, annotation for enzyme datasets of this scale is unreliable, so the labels used for this evaluation are IP signatures. The MDP was solved for K=500 as a single run for each algorithm.

For the ATF_TR dataset and for K=500, the greedy MDP solver had an IPR coverage of 0.85 and a GSI of 0.905. The TS solver had an IPR coverage of 0.77 and a GSI of 0.909. Therefore, based on IPR signatures alone, the greedy MDP solver slightly outperforms the tabu search algorithm. However, as was shown in section 4.3.4, there is not a direct correlation between IP signature diversity and EC class diversity, the latter of which is a better indicator for the purpose of diversity in catalytic profile.

Therefore, a SSN at 40% threshold was also built for ATF_TR to examine the spread of the chosen subsets in sequence space, with the tabu search subset highlighted in Figure 4.9 and the greedy subset highlighted in Figure 4.10. Both solutions show high topological spread, with nodes selected across all major and most minor components of the SSN by both solutions. The subgraph of both solutions is also expectedly less dense, going from a density of 0.026 to 0.004 for the tabu search solutions, and even lower to 0.0001 for the greedy algorithm solution.

The tabu search solution also has a significantly higher average degree of 60.42 compared to the greedy solution's average degree of 20.23, for this SSN at 40% threshold. However, the average pairwise sequence identity of the tabu search solution subset is 13.7%, which is lower than the greedy solution's 16.8% average sequence identity. It is therefore clear that while the distance between sequences is being maximised

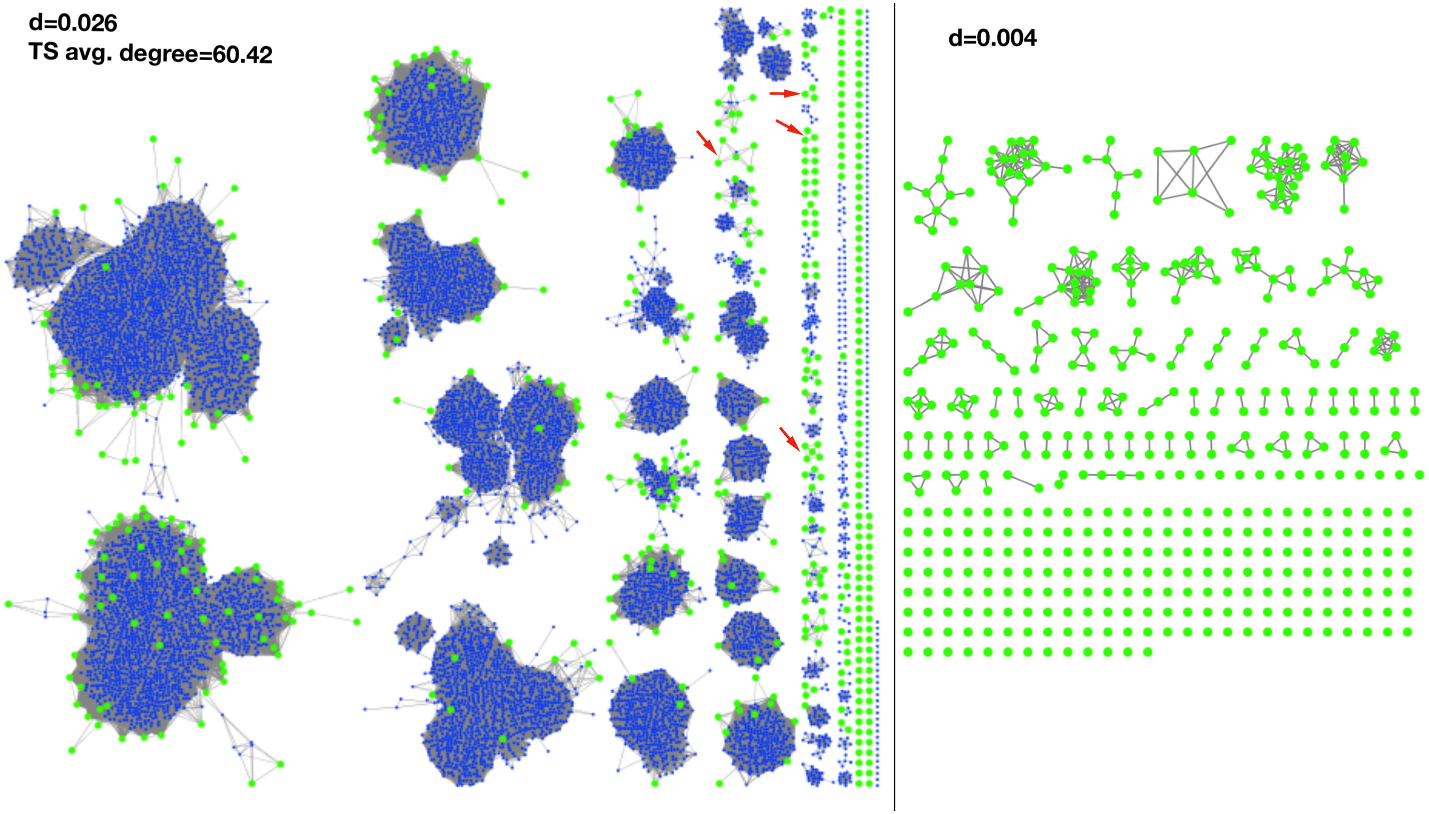


Figure 4.9: SSN at a 40% identity threshold for the ATF_TR dataset. Green nodes represent the enzymes sampled by the tabu search MDP algorithm, the subgraph of which can also be seen, along with the average node degree for the original graph. The network density for the SSN and the MDP solution subgraphs can also be seen. Similarly to the smaller datasets, the topological spread of the chosen nodes is high, and the subgraph density is lower than the graph's. The average node degree for this graph is significantly smaller than for the greedy solution seen in Figure 4.10. However, the tabu search solutions does seem to pick many nodes from small but very similar connected components (pointed at with red arrows), showing that these solutions can likely still be optimised.

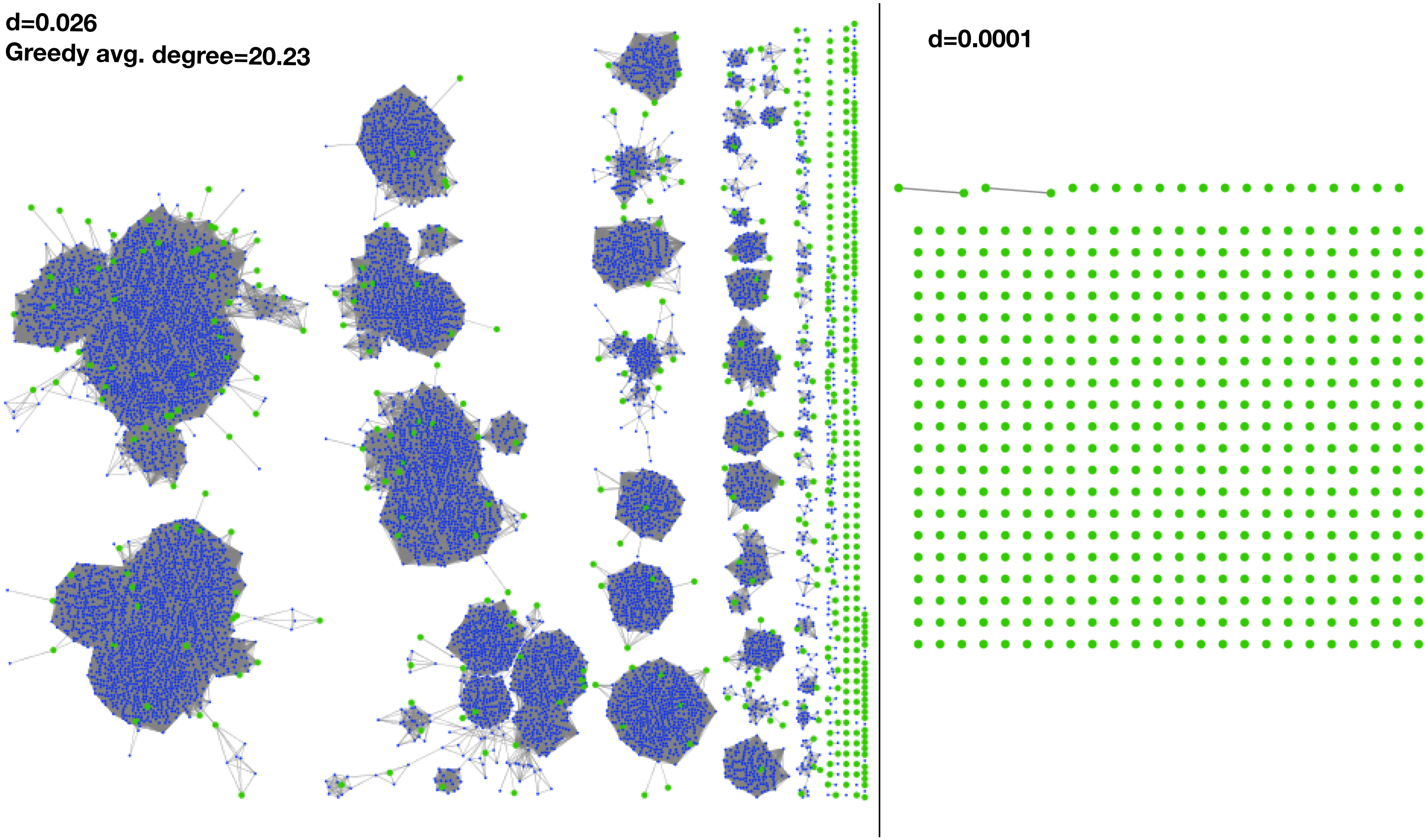


Figure 4.10: SSN at a 40% identity threshold for the ATF_TR dataset. Green nodes represent the enzymes sampled by the greedy MDP algorithm, the subgraph of which can also be seen. The network density for the SSN and the MDP solution subgraphs can also be seen, along with the average node degree of the original graph. Similarly to the smaller datasets, the topological spread of the chosen nodes is high, and the subgraph density is lower than the graph's.

by both algorithms, the two methods do so differently. Therefore, the tabu search subset still has lower average sequence identity even though its sequences are more densely connected at 40% identity. More thresholds need to be explored in detail, as at slightly lower thresholds the greedy solution might suddenly start containing more edges. However, one hypothesis for this result is that tabu search solutions contain more nodes that have significantly higher distances to the rest of the subset, compared to the greedy solution.

Another observation of note is that the tabu search solution does select multiple sequences from small connected components of high sequence similarity, even with the addition of the penalty term discussed in section 4.2.3, which are indicated by red arrows in Figure 4.9. These fully connected components being included in the solution subset undoubtedly also increased the density of the tabu search solution subgraph. Consequently, this result implies that while the tabu search solution is likely to be more diverse in a vacuum, it can clearly be optimised and refined further with regards to the penalty term introduced in section 4.2.3.

4.3.6 Evaluation of coevolution similarity-based MDP solutions

So far in this chapter, subsets created by maximising the distance in sequence identity between the sequences of the subset were explored. In Chapter 3, the use of similarity networks built on comparisons of residue-residue coevolution patterns was described, with the primary finding being that coevolution similarity can create thresholded networks that reveal structurally and functionally interesting sequence groupings in a complementary way to sequence similarity.

It was therefore of interest to see how MDP solutions created by maximising the coevolution similarity would function compared to sequence similarity. The sequence similarity matrix was replaced by a coevolution similarity matrix produced as described in section 3.2.5. The MDP was solved for the ATF dataset for K=100 for the tabu search algorithm with EC classes as labels. The results were averaged up over 50 runs.

The average EC coverage of these runs is 0.66±0.01, while the GSI is 0.78±0.003, which are significantly worse values than the sequence identity based subsets. The

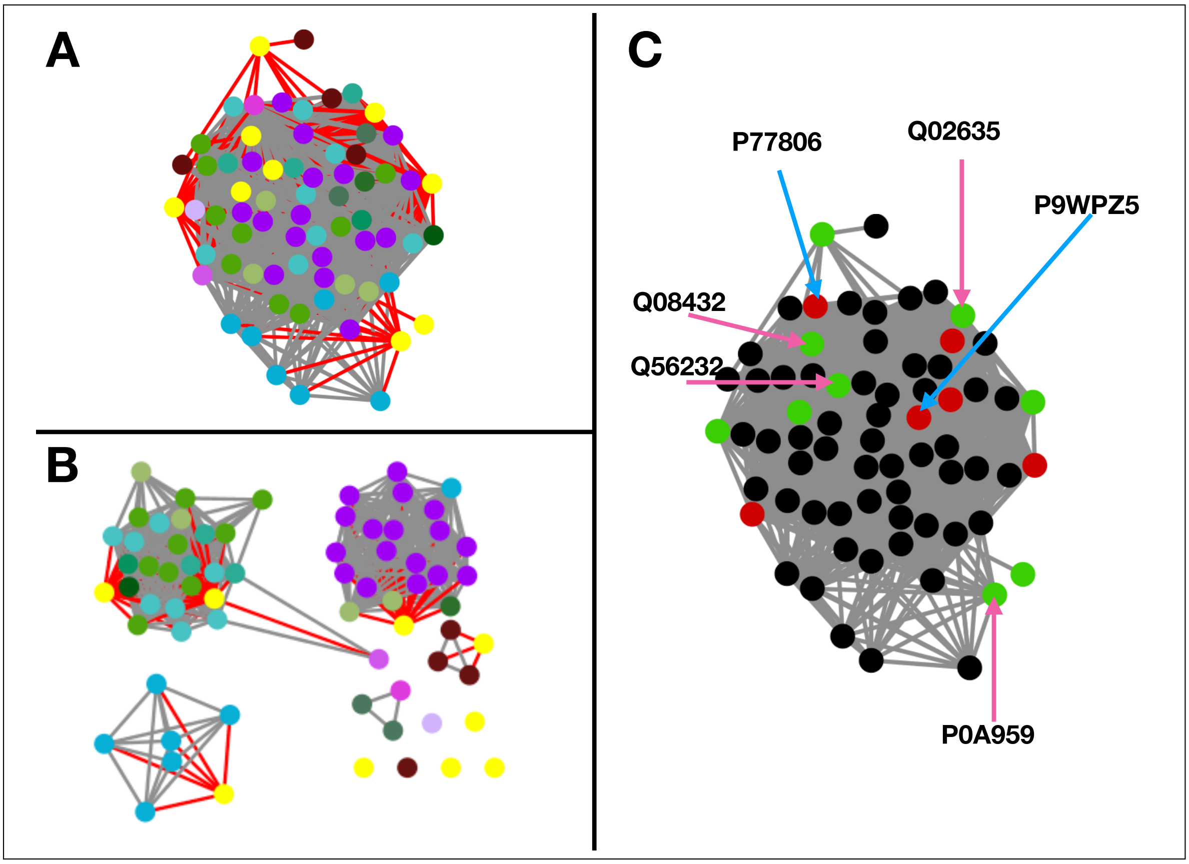


Figure 4.11: Similarity networks for the Comp69 component discussed in section 3.3.1, with nodes coloured based on EC numbers for A-B, using the same legend as Figure 3.7. The Comp69 component is a functionally diverse group of nodes that is densely connected in the CSN , but disconnected into multiple smaller components in the SSN. **A)** CSN of Comp69. Nodes highlighted in yellow are ones selected by the tabu search MDP algorithm for K equal to 100. Nine sequences of the chosen subset are located in Comp69, which cover four of the nine different EC classes contained in this component. **B)** SSN of Comp69. Nodes highlighted in yellow are ones selected by the tabu search MDP solver for K equal to 100. **C)** CSN of Comp69. Green nodes represent ones selected by the MDP algorithm, the six red nodes represent the five uncovered EC classes, and black nodes are the rest. The nodes pointed at with pink arrows are MDP-sampled nodes that have publicly available tertiary structures, while those pointed at with a blue arrow are uncovered EC classes with publicly available tertiary structures.

most likely reason for this is related to the Comp69 component discussed in section 3.3.1, which is a densely connected single component that only exists in the optimal CSN for the ATF dataset but remains disconnected in the optimal SSN (Figure 4.11, A-B). It is a very diverse component in which 9 of the 27 EC classes of the ATF dataset are unique to it.

When overlapping the sampled sequences of the highest performing run onto the op-

Table 4.8: Table showing the TM-score for the pairwise comparison of tertiary structures for the six different Comp69 nodes for which they are available. The enzymes coloured in red are those with EC classes that are not covered by the MDP solution, while green ones are covered by the MDP solution. Both P77806 and P9WPZ5 have significantly high TM-scores, and therefore high tertiary structure similarity, to multiple enzymes that are already in the MDP solution. This finding increases the likelihood that these two enzymes could be functionally similar to those already chosen by the MDP algorithm.

	P77806	**P9WPZ5**	**P0A959**	**Q02635**	**Q08432**	**Q56232**
P77806	1	0.9375	**0.84751**	**0.8871**	**0.85677**	**0.90543**
P9WPZ5	_	1	**0.87194**	**0.89878**	**0.88012**	**0.90822**
P0A959	_	_	1	0.90916	0.83243	0.89339
Q02635	_	_	_	1	0.85009	0.94061
Q08432	_	_	_	_	1	0.85941
Q56232	_	_	_	_	_	1

timal CSN, 9 of the 100 sequences chosen are located in Comp69, covering 4 different EC classes, meaning this sequence space was not ignored (Figure 4.11, A). However, of the 9 ECs that are not covered at all by the subset, 5 of them are in Comp69. Also, these 5 uncovered Comp69 labels exist on only 6 sequences total. Given the already high representation of sequences from Comp69 in the MDP solution, the algorithm is unlikely to sample more sequences from a component that is already so highly represented. Given how densely connected in the optimal CSN Comp69 is, if sequences in this component do have more functional similarity than is known from current annotation, then MDP solutions produced using coevolution data will seem to have underperformed overall.

This MDP solution was therefore explored more thoroughly, focusing on the subset of the highest performing run. Of the nine sampled sequences that are in Comp69 (Figure 4.11), four of them have a solved tertiary structure. Of the six sequences carrying the five missing labels unique to Comp69, two of them have tertiary structure. An all-vs-all analysis of their tertiary structures was therefore performed using TM-align just like was performed in section 3. The resulting TM-scores can be seen in Table 4.8, where the two proteins with missing labels are coloured in red, while the four sequences included in the solution are coloured in green.

P77806 and P9WPZ5 carry the uncovered EC classes 2.6.1.88 and 2.6.1.17, respectively. Based on the TM-score, it can be concluded that they have high tertiary structure similarity, with the value as high as 0.94. As for comparisons between them and the rest of the sequences in the table, the TM-scores range from 0.85 to 0.91. These values are especially high when compared with Q02635 and Q56232, for which they range from 0.89 to 0.91. And yet, the sequence identity between them ranges from 26.2% to 33.2%, which are relatively low. Therefore, while the annotation to support such a conclusion does not exist, these results indicate that the MDP solver not choosing more sequences from Comp69 does not constitute a worse solution in reality, due to this high structural similarity potentially also meaning high functional similarity, as was discussed in section 3.3.1.

While the coevolution-based solutions could undoubtedly be better as shown by lower values of GSI, the worse performance of the MDP solver indicates missing and/or incorrect annotation played a part. Such problems will tend to inflate sequence similarity-based results as much of the existing annotation will be automatically assigned using such data, even if incorrect. As this study only sampled sequences based on coevolution data for one dataset, it therefore is inconclusive where the such data stands relative to the sequence similarity-based results. One certainty however is that the MDP tabu search algorithm, due to its nature as a metaheuristic, can be applied with any sort of relevant similarity measure and optimally choose subsets that maximise the inherent features of said measure. This is evidenced by how the sequences in Comp69 of similar tertiary structure but lower sequence identities were not chosen in larger amounts, as coevolution similarity acts as a good proxy for tertiary structure similarity.

4.4 Discussion and conclusions

As the amount of unannotated sequence data deposited to public databases continues to increase, the optimised selection of enzymes for panels to be characterised in the laboratory becomes more important. Current methods of sampling such panels often necessitate a high level of prior knowledge and annotation about an enzyme family's overall structure, and time-consuming construction of models like phylogenetic trees and SSNs which then have to be manually interpreted. These factors are further amplified for more challenging applications like enzyme function analysis, as they require even more stringent levels of knowledge to make confident assertions.

In this work, a novel method for the subset selection of functionally diverse enzyme sequences was introduced, through the solving of the maximum diversity problem, a classic computer science problem. This method is entirely context and knowledge free, as it functions purely based on the primary sequence of enzyme proteins in a dataset.

4.4.1 Strengths and limitations of the MDP approach

The primary strength of the method shown in this work is that solving the MDP for a dataset with an all-vs-all sequence identity matrix as input results in subsets containing high catalytic diversity, as determined using three publicly sourced enzyme family datasets and their assigned functional labels. The solutions produced contain both high richness i.e. much of the known functional classes of a dataset are present in a solution subset, and high relative abundance i.e. labels present in a solution subset are generally as abundant as each other for both EC classes and IP signatures.

Furthermore, chosen proteins laid out on sequence similarity networks show high topological spread, with the majority of connected components having representative nodes being chosen. This is also evidenced quantitatively by significant reductions in network density when transforming the SSNs into their respective solution subgraphs and the relatively high proportion of nodes which are edgeless in the subgraphs. A similar observation is made when the proteins are instead laid out on a phylogenetic tree, with sampled sequences often being on the edge of sequence space.

A significant strength of the MDP method beyond the high diversity shown in the subsets is the lack of existing annotation about the dataset necessary for the it to function. As the algorithm only requires primary sequences as input, and also attempts to make up for potential sequence bias, it is able to automatically select sequences that are the most distant from each other as possible based on identity. By proxy, this method therefore produces panels of varied enzymatic profiles and specificities when applied to a dataset containing such diversity, without needing to know the actual canonical diversity present. This strength is particularly helpful when applied to large novel sets of sequences, for which the MDP method was also shown to function at a high level. Also, this method of diverse enzyme selection is purely automated, and therefore requires no manual interpretation. Therefore, the method tackles two of the limitations of current panel selection approaches described in section 2.2.3 of the Background.

One potential disadvantage of this method is that sequence bias is likely to always affect the quality of the solutions to an extent, as shown by the presence of some high identity edges in the sequence similarity subgraphs (Figure 4.9). Therefore, it would still be helpful to complement MDP solutions with manual examination and refinement of the solution to counteract any present bias. Also, preprocessing of datasets to reduce redundancy and very high identity pairs would help with this problem.

Also, the significantly better MDP solving algorithm is tabu search, which is a stochastic algorithm. For datasets with minimal or no annotation to assess the quality of subsets, it not clear how to optimise the choice of solution produced by multiple tabu search runs. However, this potential limitation is unlikely to be much of a problem given how the functional diversity of produced solutions seems to converge well, as shown by the low standard deviations of every metric assessed in this work.

Another limitation of this work lies in that the GSI as a metric becomes less useful the higher the number of overall labels exist, to the point that even randomly selected subsets can have higher values for the GSI. When using labels like InterPro signatures, this limitation becomes more challenging as most proteins will have multiple signatures at a time. This challenge was made obvious by the difficulty shown in this work in assessing relative abundance of MDP solutions when InterPro signatures were used, with a weighted GSI approach [170] likely having been better placed for such labels.

A final limitation of this work is the selection of the medoids to create solution subsets in the comparison of the MDP approach with k-medoid clustering. It is possible that sampling enzymes around each medoid might have better results. However, given the level of difference in performance between the k-medoid method and the MDP algorithm it is still unlikely to reach that level from such a change alone.

4.4.2 A de novo approach to diverse sampling from enzyme families

The tabu search method for solving the MDP is a meta-heuristic, meaning that it can solve problems independent of specification and details. This is demonstrated by how easy it was to swap in the coevolution similarity information instead of the sequence identity metric, the results of which matched up with the strengths of the former metric. Nothing about the problem specification needed to be modified, because it is a context-free approach that simply maximises a solution based on a mathematical objective function.

This method can therefore be applied *de novo* to datasets for which little to no information exists about their sequences, such as metagenomic datasets. It is possible to define new objective functions, choose different distance metrics to achieve said objective, and change how the solutions are assessed, but the result will stay the same; a subset of proteins for which the given distance metric is maximised. Such a method is powerful because annotation-based selection is made difficult due to the potential for incorrect annotation and the reality of incomplete annotation. Also, potential applications of the MDP method is not are not limited to maximising diversity in catalytic function, but could also include other enzyme attributes: diversity in structure, physiochemical properties like optimal pH and temperature, etc.

Undoubtedly, the MDP approach does not produce perfect solutions, by nature of it being a metaheuristic, but also due to the reality of enzymes being highly complex macromolecules that cannot be grouped with perfect accuracy using similarity measures alone. Therefore, an automated method like the MDP could benefit from refinement by manual curators to select areas where a generated subset could be improved or reduced in size. In particular, currently existing tools like SSNs and phylogenetic trees can help with such refinement, but also novel methods described in this thesis such as CSNs (Chapter 3). In this context, an MDP-based panel can provide a diverse initial subset that could be further optimised using CSNs.

Selected enzyme family panels could then be biochemically characterised under laboratory conditions. Given the inherent functional diversity of the selected enzyme subsets, such characterisation assays could become more time and cost effective using the methods developed in this work.

4.4.3 Future work

One avenue for future work would be the use of different identity metrics. For example, while global identity was used, BLAST identity could be a way of reducing the running time of the method provided it produced subsets of similar functional diversity. Other potentially interesting metrics include secondary structure similarity, solvent accessibility similarity, and tertiary structure similarity.

This first avenue for future work leads to a potential second: the aggregation of multiple different similarity scores. It would be interesting to see how a concatenation of all the strengths of different similarity metrics can help produce better solutions than with individual metrics. One could then perform parameter optimisation on such an aggregate score to determine which metric is most influential for the purpose at hand.

Also, as the quality of coevolution-based MDP solutions was only assessed based on one dataset, the result of that endeavour are inconclusive for now, though it performed as expected for said dataset. In the future, performing this analysis on more data would help clarify how well coevolution similarity would adapt as a functioning metric for such a sampling method.

Finally, there would be value in developing a pre-processing pipeline that assures datasets do not suffer from common pitfalls like sequence bias and sequence length outliers at both ends.

4.4.4 Conclusions

Solving the maximum diversity problem using sequence similarity as a metric appears to produce functionally and phylogenetically diverse subsets based on gold-standard datasets and labels. This method outperforms standard clustering methods like k-medoids, and can also be applied to larger datasets without a drop in performance. MDP-based panel construction can therefore function as a context-free technique for automatically generating enzyme family panels of a desired size.

5

Autoencoder-generated sequences for artificially increasing enzyme family diversity

Contents

5.1 Introduction

In the previous two chapters, two novel methods were described - Chapter 3 introduced a novel network method for the functional analysis of enzyme families, and Chapter 4 described a novel technique for automatically sampling subsets of enzymes that are catalytically diverse from a larger superset. These two complementary methods focus on gaining a better understanding of existing enzyme datasets, either by visualising clusters in a family for the former, or through the automatic sampling of enzymes that are representative of the family's diversity for the latter. The methods of the previous two chapters therefore facilitate the selection of putative enzymes from larger datasets. This selection process aims to optimise enzyme panel creation for laboratory-based characterisation, so that enzymes of novel properties can be discovered and used by industry.

However, another approach to increase the portfolio of useful biocatalysts is the engineering of already discovered and well-characterised enzymes. In particular, the engineering of enzymes has been successful in artificially introducing novel functions and properties through changes to their amino acid sequence. In these experiments, an enzyme is mutated to reach a state with novel properties and functions [171, 172]. While the resulting enzymes are artificial, such modifications have been shown to be successful, including engineering enzymes to be resistant to chemical oxidation [173], increasing stability against certain solvents [174], and for increasing catalytic activity [175].

However, while it is possible to create synthetic enzymes that are functionally viable, enzyme engineering is often very manual, complex, and requires much trial-and-error in the laboratory to be successful. While the generation of synthetic enzymes offers an avenue for increasing the diversity present in enzyme families, *in silico* approaches that optimise the creation of such enzymes, *de novo*, would be valuable. As described in section 2.4 of the Background, machine learning has been at the forefront of multiple recent breakthroughs in bioinformatics. This chapter describes research into the development of a machine learning-based method that attempts to achieve this goal.

5.1.1 Neural network applications for generating novel sequences *de novo*

Recent developments in the *de novo* generation of enzyme sequences mainly consists of the use of deep learning neural networks [176–179]. For example, Wan and Jones [177] used a generative adversarial network (GAN), which is a type of neural network, to produce synthetic sequences that help improve the performance of Gene Ontology (GO) term prediction tools. Repecka and colleagues [178] trained another GAN to create 55 novel enzyme variants of malate dehydrogenases, of which 13 were successful in the laboratory both in terms of solubility and catalytic activity.

Neural networks are popular for such applications because they have been proven to be highly successful at learning and recognising subtle features of a given training dataset i.e. recognising snout differences between a dog and a cat to label a picture as containing either one of the latter two. Protein sequences also contain features, such as secondary structures and solvent accessibility, which can be used in classification. These features make them highly relevant for models like neural networks for recognition and prediction. Indeed, many of the state-of-the-art structural bioinformatics tools that predict such features use neural networks, like PSIPRED for secondary structure prediction [180], MetaPSICOV for contact-map prediction [181]. Also, even enzyme-specific prediction problems have successful machine learning-based methods, such as the prediction of the Enzyme Commission number (EC) using DEEPRe [88] and active site detection [182]. Therefore, neural networks that learn such important enzyme features for prediction purposes, can also learn them for generative purposes, as shown by Repecka and colleagues [178].

5.1.2 The logic of autoencoders

As discussed in the preceding section, neural networks are able to learn complex features inherent to protein sequences [88, 181]. Also, neural networks have been successful at generating novel enzyme sequences *de novo* that are functionally viable [178]. Therefore, neural networks are a promising avenue for attempting to artificially increase the catalytic diversity of an enzyme family using a generative approach.

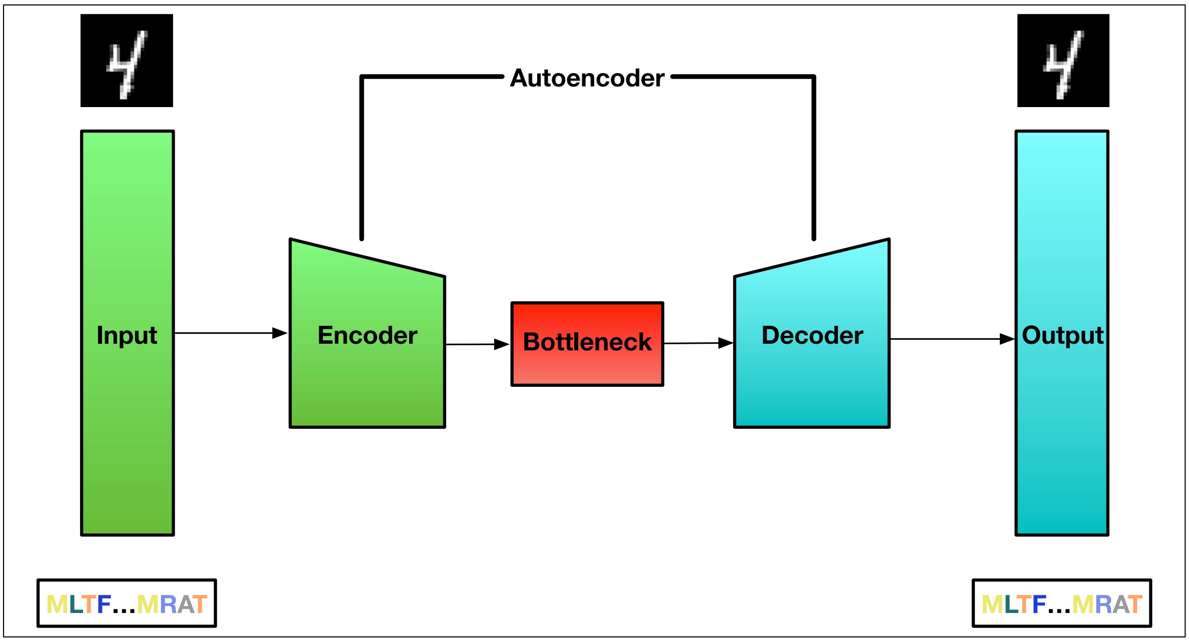


Figure 5.1: Diagram representing the logic of an autoencoder neural network.This encoder block of layers compresses the input into the bottleneck or latent space layer. The bottleneck is a compressed vector that should contain the inherent information of an input summarised as a smaller set of features. Then, the bottleneck is passed to the decoder, which attempts to decode the latent space layer back into the origin input. Autoencoders have been shown to learn to reconstruct various inputs, from images of numbers to amino acid primary sequences.

More specifically, one neural network structure of interest that was used in this work is called an autoencoder [176, 183]. Autoencoders are made up of three main layers: an encoder, a decoder, and a bottleneck (Figure 5.1). Autoencoders work by first condensing some input into a smaller encoded representation called a bottleneck through the encoder layer. The bottleneck (also called a latent space) is then usually decoded back into the original input using the decoder layer.

The logic of autoencoders lie in their dimensionality reduction properties. As an example, an autoencoder could be given the image of the number four as input (Figure 5.1). When the image is passed to the encoder, it will be mathematically transformed into a condensed vector summarising the input. The decoder then attempts to reconstruct the number four from the bottleneck representation during the training process. Consequently, an autoencoder learns how to encode an input into the condensed bottleneck into a shape that is amenable to being decoded back into its original form.

This training logic results in the bottleneck vector consisting of important inherent features that summarise the input - in the case of the number four, it could be the different corners and edges that make up the number. When applied to protein primary sequence (Figure 5.1), a hypothesis can be formulated: a trained autoencoder can learn complex inherent features that make up proteins, such as secondary structures, and substrate binding sites. As a result, protein sequences encoded as such could then be summarised as a vector made up of important features [184].

5.1.3 Aim of this work

A crucial feature of an autoencoder is that they can be used to generate new synthetic data through the sampling of their latent space. Therefore, an autoencoder trained to reconstruct proteins could then be used to generate completely novel and artificial sequences. As such, a network can be trained on native protein sequences, from which learned features from the bottleneck can then be sampled from to be decoded into sequences resembling viable proteins [176, 179].

While Costello and and Martin [184] have trained an autoencoder to generate synthetic enzymes in the manner just described, they have not studied in significant detail the viability of such synthetic enzymes in the laboratory. Also, while Repecka and colleagues [178] have shown that using neural networks can produce functionally viable enzymes, their results were limited to a specific use case of synthetic malate dehydrogenases, rather than an entire enzyme family. Therefore, to the best of the knowledge of the author, no study on the use of autoencoder-generated enzyme sequences for increasing the amount of catalytic diversity in an enzyme family has yet been performed.

The principle aim of this work was therefore to research a method for generating additional catalytic diversity in an enzyme family by sampling novel primary sequences from the latent space of a trained autoencoder model. This aim was separated into two primary objectives:

1. A discriminant autoencoder was designed and trained on a large set of curated enzyme sequences, along with an assessment of the reconstruction quality.

2. A pipeline to sample and select novel enzyme sequences from the autoencoder

bottleneck layer was developed, along with a thorough study of how functionally viable sampled sequences might be.

The second objective of this work resulted in the curating of 30 synthetic enzyme sequences from the aldo-keto reductase (AKR) family, which were then assessed in the laboratory in terms of overexpression and solubility.

Attribution: *The laboratory experiments carried out on the set of synthetic sequences were performed by collaborators at Prozomix Limited. The methods described in section 5.2.7 and the SDS-PAGE results in section 5.3.5 are therefore fully attributed to them.*

5.2 Methods

5.2.1 The datasets

As described in section 2.4 of the Background, to perform machine learning, two datasets are needed:

1. A training dataset, which is used to train the neural network's weights to solve a well-defined problem.
2. A testing dataset, which is used to assess the performance of the neural network on unseen data.

The training dataset was built by retrieving all enzyme primary sequences on Swiss-Prot [128, 185]. Specifically, the SPARQL endpoint of Swiss-Prot was queried:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX up:<http://purl.uniprot.org/core/>

SELECT DISTINCT ?prot ?enz ?aa_sequence
WHERE
{{
        ?prot a up:Protein .
        ?prot up:reviewed ?status .
        ?prot up:sequence ?seq .
        ?seq a up:Simple_Sequence .
        ?seq rdf:value ?aa_sequence .
        ?prot up:enzyme ?enz .
        FILTER ( ?status IN ( true ) )
}}
```

From the resulting enzyme set, sequences with more than 800 amino acids were discarded to remove potential multifunctional enzymes. Enzyme sequences containing any ambiguous amino acid positions (B, Z, X, J) or any of the non-primary 20 amino acids (O, U) were also discarded. The testing dataset was made up of 5,309 enzyme sequences that were retrieved from TrEMBL instead of Swiss-Prot, as all the enzymes on Swiss-Prot were included in the training dataset. This choice of query database was made to better guarantee that enzymes in the testing dataset are unseen data. Statistics on sequence length for both of these datasets can be seen in Table 5.1.

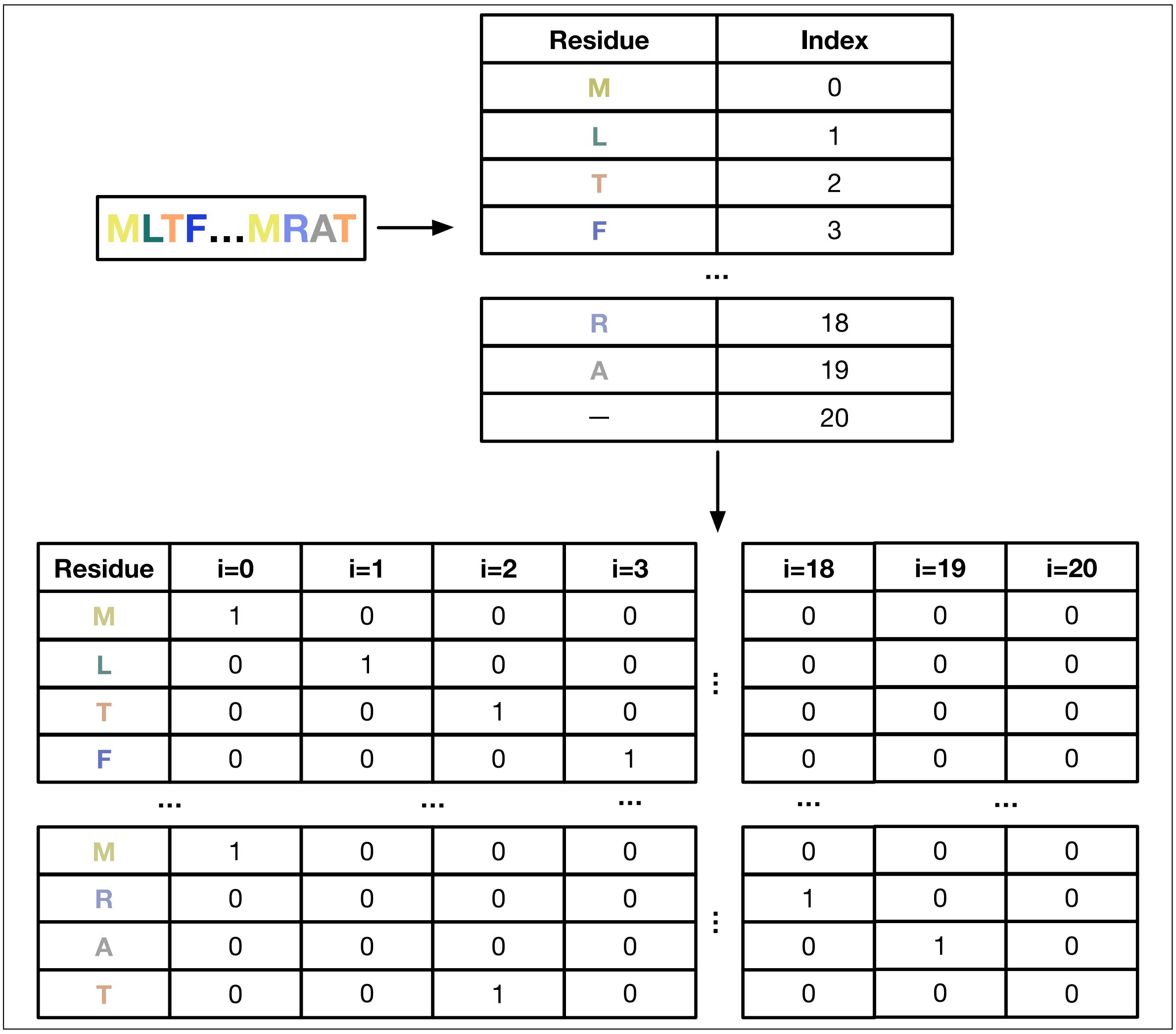
5.2.2 Discriminant autoencoder architecture

The architecture of any neural network starts with an input layer to which each sample (i.e. primary sequence) is fed. As discussed in section 2.4 of the Background, the input layer of a neural network contains a mathematical representation of the data that is amenable for a neural network to learn from. As the primary sequences in the training and testing datasets are at most 800 residues long, the length of the vector was set to 800. Each of the 800 positions - i.e. residues - of this vector can be one of the 20 different standard amino acids. Each position can also be a blank position to represent sequences of lengths shorter than 800. The width of the input vector is therefore 21, with a value of 1 for the specified residue, for a total vector dimension input of 800x21. This representation of sequences is called a one-hot encoding and is standard for such data, an example of which can be seen in Figure 5.2.

The rest of the neural network architecture can be seen in Figure 5.3. Following the input layer are two blocks of 1D convolution layers and maxpooling layers. These layers

Table 5.1: Table with summary statistics about the training and testing datasets. Q1 to Q3 represent the sequence lengths of each quartile.

Dataset	Number of sequences	Average length	Q1 Length	Q2 Length	Q3 Length
Training	244,514	351.5	240	334	443
Testing	5,309	332.9	213	308	428



Residue	Index
M	0
L	1
T	2
F	3
...	
R	18
A	19
—	20

Residue	i=0	i=1	i=2	i=3		i=18	i=19	i=20
M	1	0	0	0		0	0	0
L	0	1	0	0	⋮	0	0	0
T	0	0	1	0		0	0	0
F	0	0	0	1		0	0	0
...		...		...		...		...
M	1	0	0	0		0	0	0
R	0	0	0	0	⋮	1	0	0
A	0	0	0	0		0	1	0
T	0	0	1	0		0	0	0

Figure 5.2: An example of one-hot encoding using amino acid sequence. In one-hot encoding, a sequence of characters of size L is transformed into a vector of size $L * C$, where C is the size of the alphabet. In the case of amino acid sequence, the alphabet size is 21, with 20 different amino acids plus one to represent blank positions. Each letter (or residue) of this alphabet has an index, and each individual character of a sequence is given a value of 1 for their respective index, and 0 for all others.

help summarise and condense the input from a latent space vector with dimensions of 800x21 to one with dimensions 200x32 - a 61% decrease in vector size. This section of the autoencoder is effectively the encoder.

Next, the autoencoder splits into two branches - a reconstruction branch, and a discriminant branch. The reconstruction branch scales the bottleneck layer back to an output of the original vector size of 800x21, attempting to decode the bottleneck back into the original input.

The discriminant branch flattens the bottleneck into a one-dimensional vector, which is then connected to a single output node. As will be discussed in section 5.2.3, the training process includes randomly generated primary sequences, which the discriminant will try to differentiate from the real sequences. The purpose of the discriminant branch is to teach the neural network ”what not to know”, to get as close as possible to a latent space that truly learns important features of enzyme sequences. The use of this discriminant turns the neural network into a discriminant autoencoder.

5.2.3 The training process

The neural network described in the preceding section is implemented using Keras [4], a Python library that helps abstract the creation and training of neural networks. The Python Keras code that generates and compiles the discriminant autoencoder model is the following:

```
def create_discriminant_autoencoder():

        # Input -> Encoded representation
        input = Input(shape=(800, 21))
        nn = Conv1D(64, (5,), activation='relu', padding='same')(input)
        nn = MaxPooling1D((2,), padding='same')(nn)
        nn = Conv1D(32, (5,), activation='relu', padding='same')(nn)
        encoded = MaxPooling1D((2,), padding='same')(nn)

        # Encoded -> Decoded Representation
        nn = Conv1D(32, (5,), activation='relu',
         ↪   padding='same')(encoded)
        nn = UpSampling1D(2)(nn)
        nn = Conv1D(64, (5,), activation='relu', padding='same')(nn)
        nn = UpSampling1D(2)(nn)

        # Output
        decoded = Conv1D(21, (5,), activation='softmax',
         ↪   padding='same', name='decoder')(nn)
        flat = Flatten()(encoded)
        discriminant = Dense(1, activation='sigmoid',
         ↪   name='discriminant')(flat)
        outputs = [decoded, discriminant]
        losses = ['categorical_crossentropy', 'binary_crossentropy']

        # Compile the model
        autoencoder = Model(inputs=input, outputs=outputs)
        autoencoder.compile(loss=losses, optimizer=Adam())
```

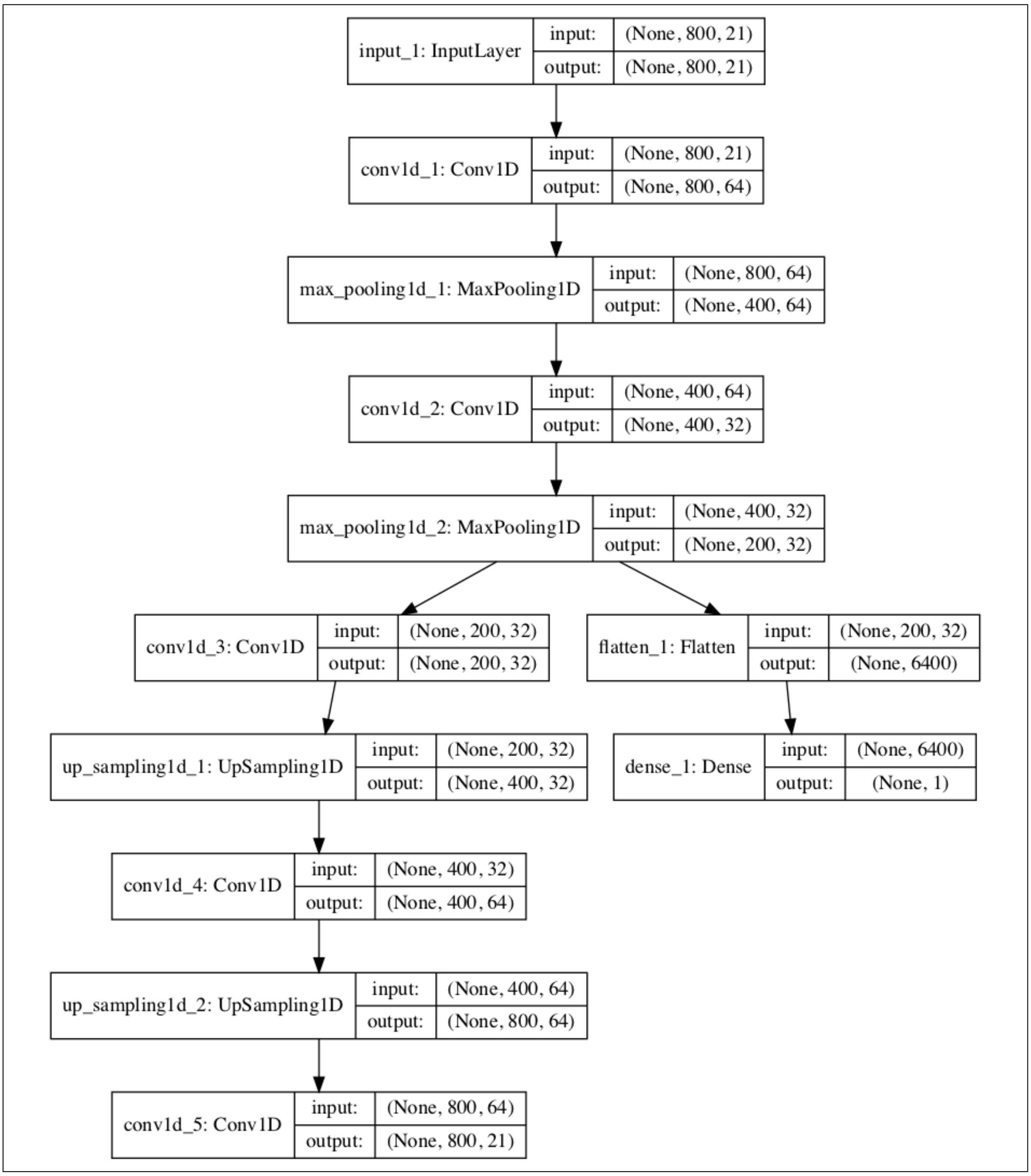


Figure 5.3: Neural network architecture of the discriminant autoencoder implemented for this work, produced using Keras [4]. The first layer is the input layer, followed by two blocks of convolutional layers that condense the input into the bottleneck. Then, the decoder branch on the left performs the inverse transformation, upsampling the bottleneck back to the original dimensions of the input. Finally, the discriminant branch to the right helps teach the autoencoder what not to learn through the use of random sequences.

As described in background section 2.4 of the Background, neural networks are highly parametrised models. These parameters were optimized through a systematic manual exploration and the values can be seen in Table C.1 of the Appendix.

Since the network is a discriminant autoencoder, randomly generated sequence data was added to the training dataset for the discriminant to learn from, to teach the neural network ”what not to know”. This was done by creating amino acid sequences that start with a methionine, followed by a random assortment of the 20 amino acids, for a random sequence length between 150 and 800 residues.

The training process was then as follows: at every epoch, the training dataset is shuffled along with the concatenation of 5000 randomly generated sequences. Then, for every epoch, the training data is input in batches of 16 sequences, which is a standard batch size number. This occurs for five epochs, with newly generated random sequences at every iteration.

To assess the performance of the training process, after five epochs all the sequences of the training and testing datasets described in section 5.1 are passed through the autoencoder. Then, the reconstructed sequence is aligned to its respective original sequence using global Needleman-Wunsch pairwise alignment [65]. The average training and testing pairwise identities were finally produced as a result.

5.2.4 Generation of novel enzyme sequences

Novel synthetic enzymes were generated from the autoencoder using the same method as Costello and Martin [184], which is made up of steps 1 to 3 Figure 5.5. A template set of enzyme sequences as input was chosen and then encoded into the latent space vector. Then, the mean and covariance matrix of the latent space vector was computed. Using the mean and covariance matrix of the template, a Gaussian distribution was then modelled and sampled from. Specifically, each sample represents a bottleneck vector representing the learned latent space. Each of these vectors were then finally decoded, creating a cohort of novel and synthetic primary sequences.

The template enzyme sequences were chosen from the AKR family, as they are highly diverse biocatalysts of interest to industry. AKRs have the PFAM id PF00248, which

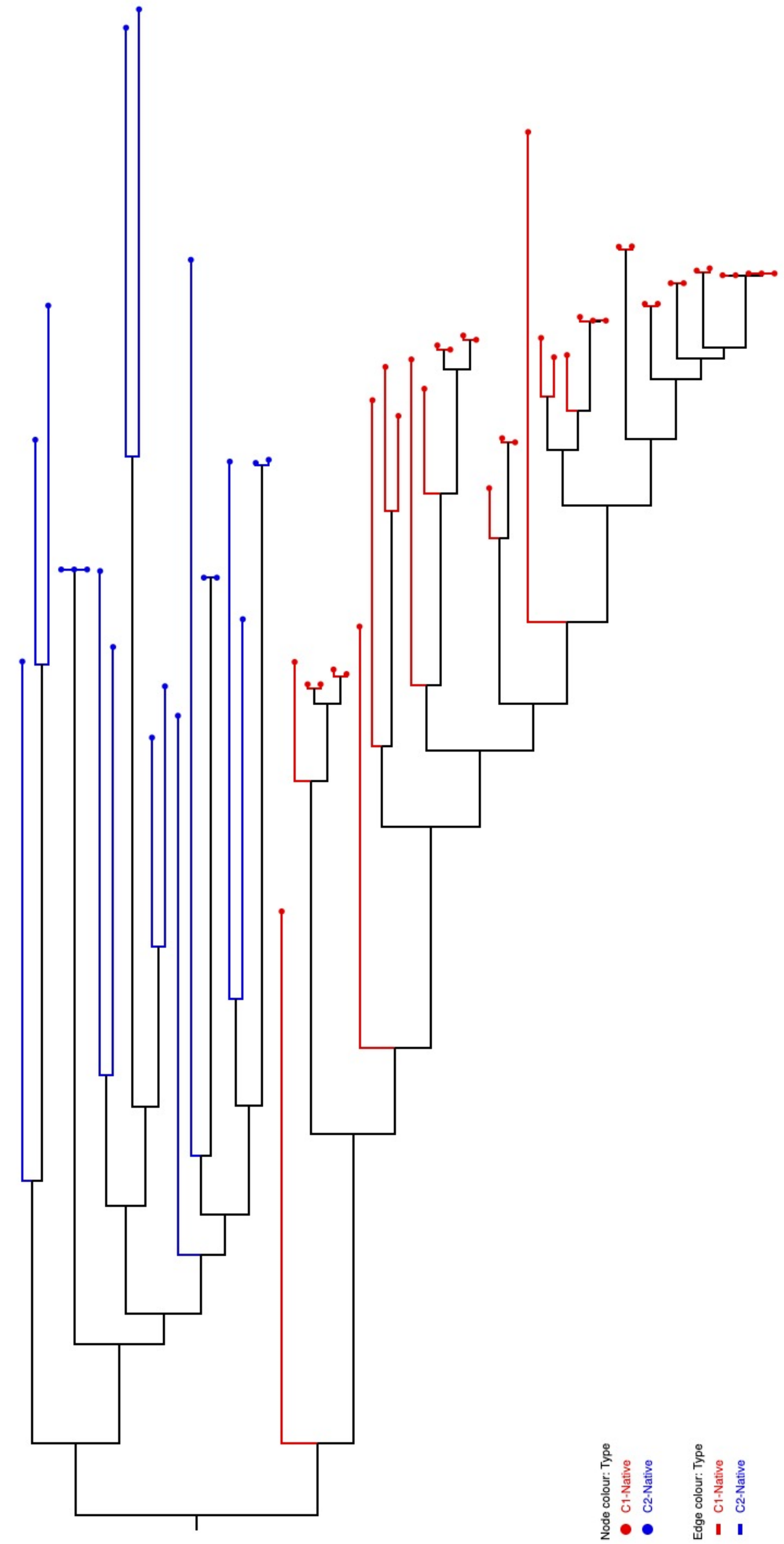


Figure 5.4: Phylogenetic tree of the AKR59 template dataset. Red leaves are the 39 Clade1 enzymes, while blue leaves are the 20 Clade2 enzymes.

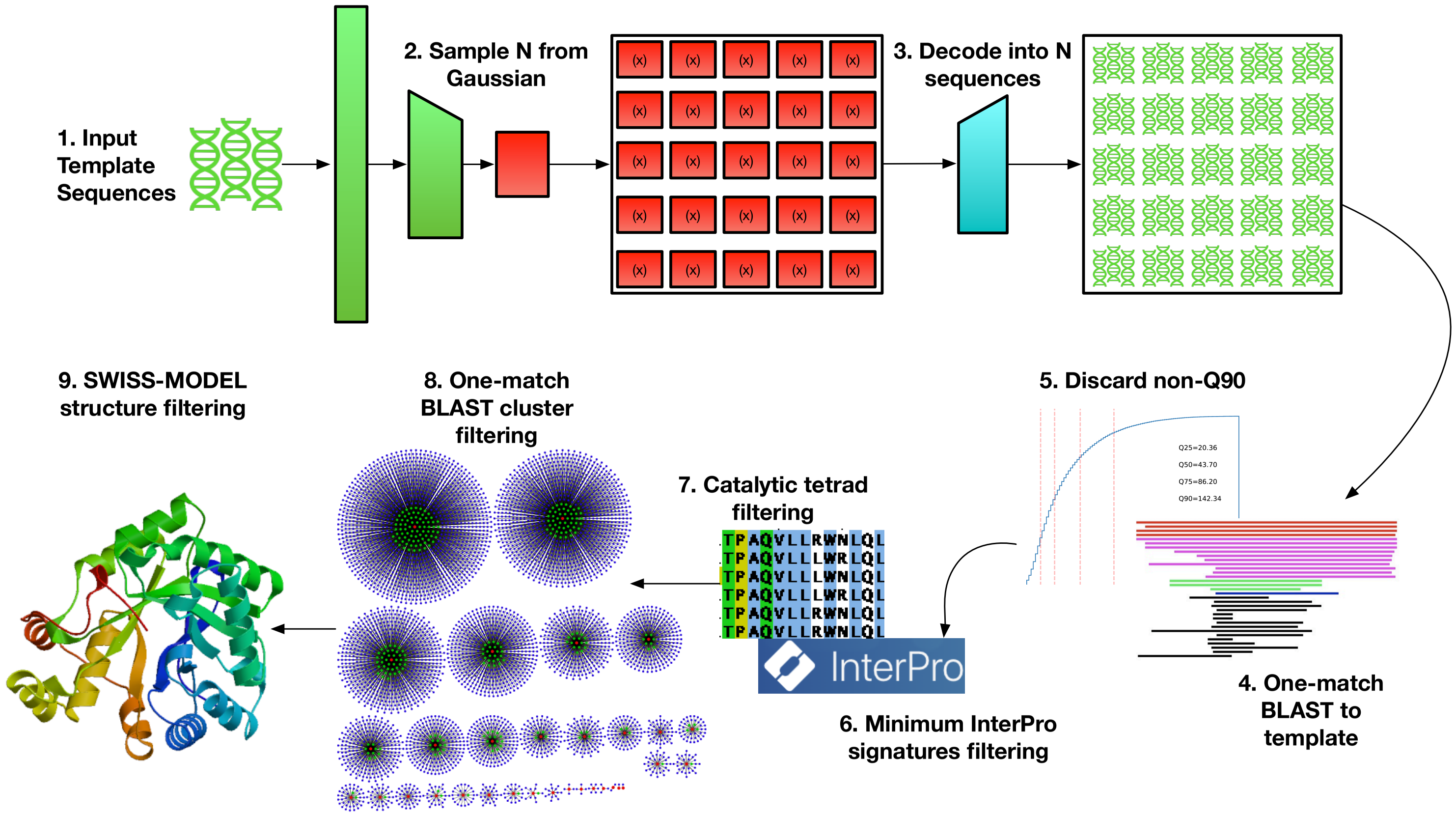

1. Input Template Sequences
2. Sample N from Gaussian
(x)
3. Decode into N sequences
5. Discard non-Q90
Q25=20.36
Q50=43.70
Q75=86.20
Q90=142.34
4. One-match BLAST to template
6. Minimum InterPro signatures filtering
InterPro
7. Catalytic tetrad filtering
TPAQVLLRWNLQL
TPAQVLLWRLQL
TPAQVLLWNLQL
TPAQVLLWRLQL
TPAQVLLRWNLQL
TPAQVLLRWNLQL
8. One-match BLAST cluster filtering
9. SWISS-MODEL structure filtering

Figure 5.5: The workflow for the generation (steps 1-3) and filtering of synthetic enzymes utilised in this work (steps 4-9). **1.** A set of template sequences representing the enzyme family of interest was used as input into the encoder portion of the autoencoder. **2.** The latent space of N sequences were sampled from a gaussian distribution using the mean and covariace matrix of the latent space that encodes the template set. **3.** The N sequences were input into the decoder part of the autoencoder to generate novel sequences. **4.** The N sequences were 'one-match' BLAST-ed to the template sequences, where only the best match was recorded at an e-value threshold of $1 * 10^{-10}$. Sequences without a match were discarded. **5.** The 90^{th} percentile of -log(e-val) was computed, and synthetic sequences with a one-match below this number were discarded. **6.** InterProScan was run on the remaining sequences. Those without a user-specified set of InterPro signatures were discarded. **7.** An MSA of the template sequences and the remaining synthetic sequences were created. Those that do not conserve a set of user-specified residues were discarded. **8.** One-match BLAST similarity networks were created, where nodes are either template or synthetic sequences. Edges were made between synthetic sequences and their one-match. Sequences were sampled from each produced cluster for validation in the final step. **9.** Tertiary structure models were predicted for each synthetic sequence using SWISS-MODEL. The sequences were then ranked based on QMEAN score, with the top ones being selected.

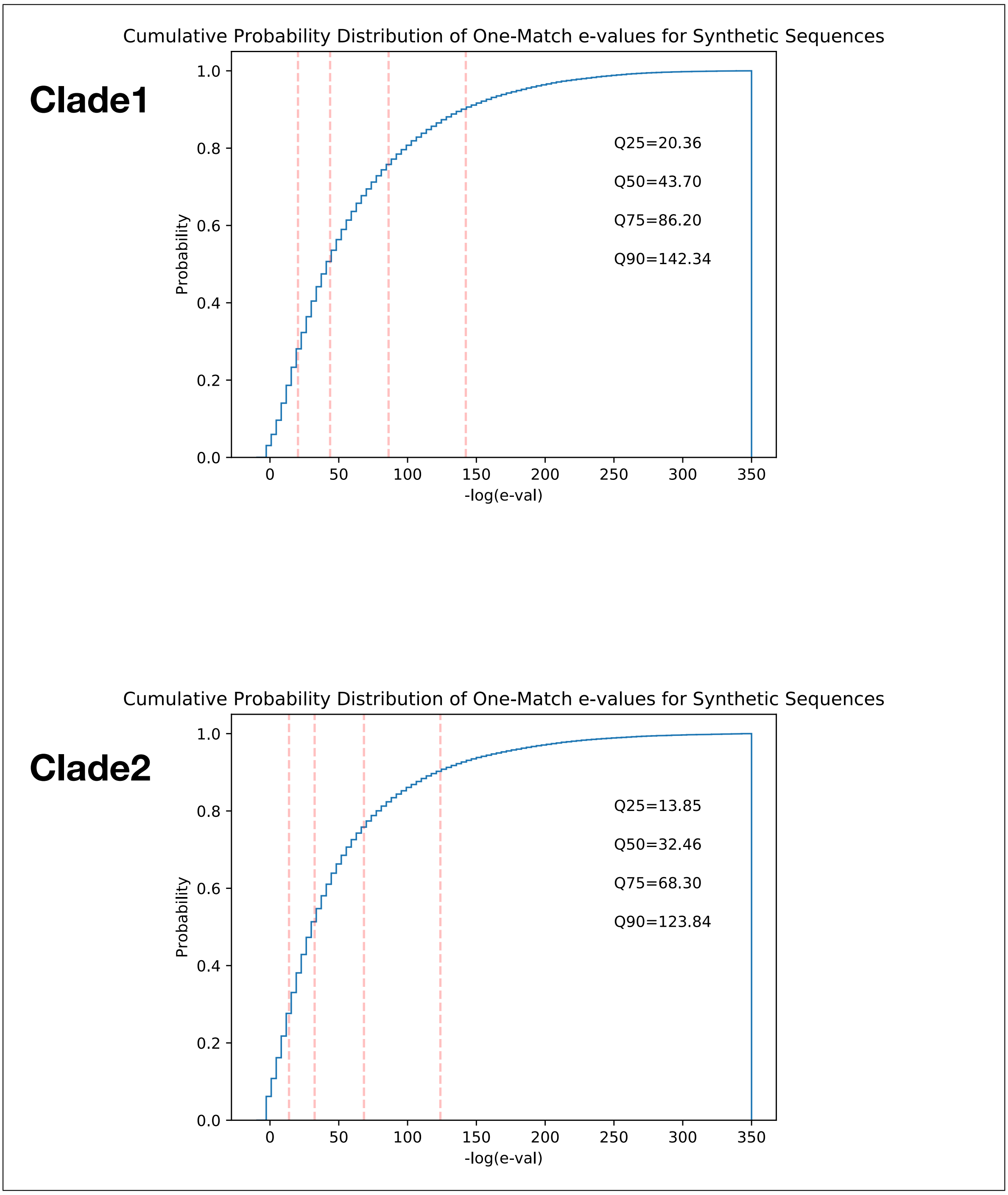


Figure 5.6: Cumulative probability distribution plots of the e-value of one-match hits. Dashed red lines are the Q25, Q50, Q75, Q90 (90^{th} percentile) thresholds. Synthetic sequences below the Q90 value were discarded for both clades.

was used to retrieve all of the prokaryotic AKRs available on Swiss-Prot. The resulting dataset contained 59 enzymes, which will be referred to as AKR59 henceforth.

An approximately-maximum-likelihood phylogenetic tree for AKR59 was produced using ClustalOmega [67] and FastTree [155], which can be seen in Figure 5.4. It can be seen that the tree contains two major clades, one in red (Clade1) and one in blue (Clade2). Clade1 contains 39 enzymes, while Clade2 contains 20 enzymes. Therefore, to produce a panel of synthetic enzymes that is representative of the AKR family, sequences were generated using each clade as individual templates rather than using all of AKR59 as one template. For each clade, 50,000 synthetic sequences were generated, for a total of 100,000.

5.2.5 The filtering pipeline

To better assess the viability of synthetic AKRs produced as described in the preceeding section, it is necessary to identify and filter those most likely to be functionally viable enzymes. This need is especially true for testing these enzymes under laboratory conditions. This filtering process can be seen in Figure 5.5 from step 4 to 9.

In the first step of the filtering, step 4, a BLAST search (at an e-value threshold of $1e^{-10}$) was performed for every synthetic sequence to its respective clade template set. The best scoring match in terms of e-value is kept and recorded. In this work, this procedure is referred to as 'one-match BLAST'. This was done by first making a BLAST database out of each template clade using *makeblastdb* [40, 64], and then running BLAST using each synthetic sequence as a query against this database, recording the best match. Enzyme sequences with no matches were immediately discarded.

Then for step 5, the distribution of e-values for all one-match hits was computed, and all sequences below the 90^{th} (Q90) percentile were discarded. This step helps better guarantee that the remaining enzymes have a core similarity with the sequences in AKR59. Steps 4 and 5 of the filtering pipeline reduce the number of enzyme sequences 50,000 for each clade to 5,090 and 5,251 for Clade1 and Clade2, respectively. The cumulative probability distribution of the e-values can be seen in Figure 5.6.

Step 6 consists of identifying the minimal set of InterPro signatures the enzymes of

Table 5.2: Table describing the four different InterPro signatures contained in the minimum set of signatures for the AKR59 enzymes. All four of these InterPro signatures are present in all template enzymes for both clades, and are used to filter out synthetic sequences that are unlikely to be viable.

Signature	Description	Database
IPR023210-PF00248	Aldo/keto reductase family	PFAM
IPR023210-cd06660	Catalytic tetrad	CDD
IPR036812-G3DSA:3.20.20.100	NADP-dependent oxidoreductase domain	CATH-Gene3D
IPR036812-SSF51430	NAD(P)-linked oxidoreductase	SUPERFAMILY

AKR59 have, and filtering out synthetic enzymes not fully containing this set. This step was performed to guarantee that the chosen synthetic sequences have the minimum sequence features that all AKRs are expected to have. The InterPro signatures for the synthetic sequences were computed using InterProScan ([54]). The minimum set of InterPro signatures for the AKR59 enzymes was then extracted, and consists of four different signatures from four different databases (Table 5.2). After this step 719 sequences remained for Clade1 and 214 for Clade 2.

In step 7, an MSA was produced containing both the remaining synthetic sequences and AKR59. A catalytic tetrad made up of four functionally important residues is known to be conserved in almost 100% of the AKR59 sequences. Using the Clade1 sequence P74308 as reference, these are D-52, Y-57, K-86, and H-119. In an effort to guarantee the presence of the correct catalytic residues in the synthetic enzymes, step 7 discards synthetic sequences that do not have all four of these residues conserved when examined on the MSA. After this step, there are only 280 and 23 synthetic sequences remaining for Clade1 and Clade2, respectively.

For the next step of the filtering pipeline (step 8), one-match BLAST similarity networks were produced, where nodes are either AKR59 enzymes or synthetic enzymes, and edges are made between the latter and their one-match template sequences. As each synthetic node has only one edge, connected components form on this network, with template enzymes as hubs (Figure 5.7).

Finally, for step 9, the synthetic enzyme with the best e-value was selected from each connected component and then passed to the homology modelling web-interface, SWISS-MODEL [78]. The purpose of this step is to assess how likely a synthetic enzyme sequence is to fold by modelling their tertiary structure. SWISS-MODEL

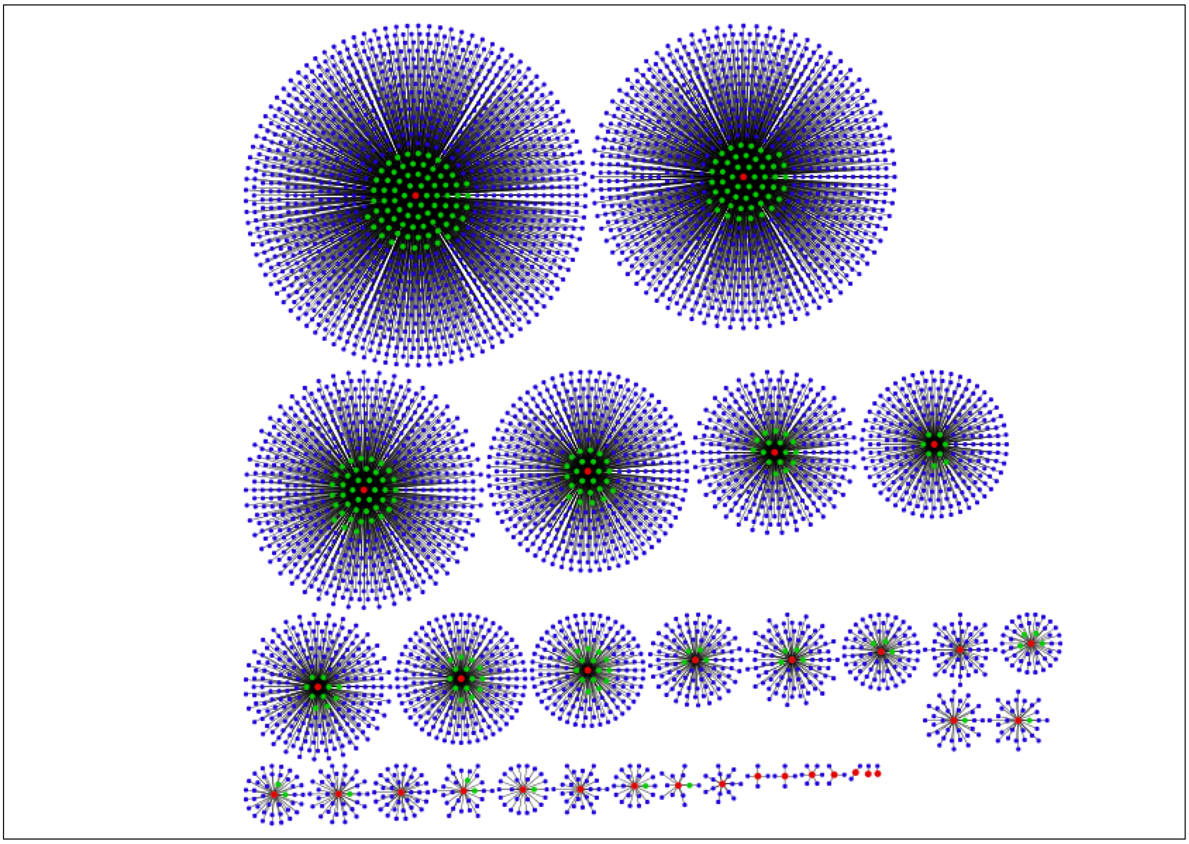


Figure 5.7: One-match BLAST similarity network, where nodes are either AKR59 (red nodes) or synthetic sequences (blue nodes), and edges are made between the latter and their respective one-matches. The top hit of each cluster was selected for assessment by SWISS-MODEL, each of which was ranked based on QMEAN score. The top 15 of each clade were kept.

produces an aggregate score called the QMEAN, with a score higher than -4.00 being considered a statistically good model. Finally, synthetic enzymes passed through SWISS-MODEL are ranked and selected based on their QMEAN scores. From each clade, 15 synthetic enzymes were selected, for a total size of 30 synthetic AKRs. This set is referred to as SynthAKR30 henceforth.

5.2.6 Assessment of predicted sequences

The performance of the discriminant branch in distinguishing between native and random sequences can provide information on whether the black box neural network model used has learned features inherent to enzymes. This performance was therefore assessed by sampling 100 random enzyme sequences from Swiss-Prot at 50 residue intervals from a range of 150 to 800. For example, 100 sequences of random length were chosen from the range of 150-200 residues, another 100 sequences were chosen from the range 200-250, and so on. Then, for each of these sequences, a random sequence of the same length was generated. Both the native and random sequences are then passed to the trained autoencoder, and the prediction of the discriminant is retrieved. A truth table analysis was then performed with these definitions:

- True positives: when a native sequence is predicted as native.
- False positive: when a random sequence is predicted as native
- True negative: when a random sequence is predicted as random
- False negative: when a random sequence is predicted as native

Three performance measures were recorded: the precision, the recall, and the F1-score. The precision helps interpret how confident the neural network is when it predicting sequences as native, while the recall helps interpret how many of the total native sequences the neural network is able to recover. The F1-score is the harmonic mean of the precision and recall, providing an aggregate score.

To visualise the similarity in the latent space between native and predicted sequences, two dimensionality-reduction techniques were used: principal component analysis (PCA) and t-distributed stochastic neighbour embedding (t-SNE). PCA and t-SNE plots were produced to compare the sequences of the training dataset with their reconstructed sequences after decoding. These techniques were also used to compare the AKR59 and SynthAKR30 sequences. Sequence similarity networks (SSN) were also generated for the union of the latter two sets, which are networks where nodes are sequences and edges are made between nodes sharing a pairwise global identity higher than some set threshold.

To assess the potential viability of the synthetic AKRs, multiple *in silico* tools were used. First, DEEPre, which is a machine learning tool that predicts EC class for enzyme sequences, was used to provide functional labels to the synthetic AKRs. These annotations were looked at in-depth. In a similar vein, the metaserver I-TASSER was used on the synthetic sequence with the most significant e-value (due to I-TASSER being computationally slow as discussed in section 2.3.3) to get an in-depth analysis of the quantitatively best synthetic enzyme generated.

Finally, detection of low complexity regions low complexity regions (LCRs) was performed on the SynthAKR30 sequences to assess whether the synthetic sequences have LCRs that align with what would be expected of working AKRs. More specifically, the prevalence of simple perfect repeats (SPRs) of amino acids, as defined by Luo and Harm [186], from repeat sizes 2 to 4 were computed for SynthAKR30 and AKR59. SPRs are repeats where the same amino acid is repeated in a sequence 2 or more times in a row. The prevalence was calculated using two equations: $\bar{R}_{aa,C}$ (Equation 1), which is the individual amino acid repeat frequency average for some set of sequences C, and $\bar{R}_C$ (Equation 2), which is the per clade amino acid repeat frequency average. For aa being a perfect repeat of one of the 20 amino acids, the $\bar{R}_{aa,C}$ simply sums up the number of occurrences of said repeat $|R_{aa}|$ in the sequences of set C, and then averages them up. This metric is calculated per amino acid repeat. $\bar{R}_C$ on the other hand is the average number of perfect repeats per sequence for all such repeats, for one of the two clades. Per perfect repeat length from 2 to 4, the $\bar{R}_C$ sums up the amino acid repeat frequency average for the sequences of a clade.

$$\bar{R}_{aa,C} = \frac{\sum_{i=0}^{C} |R_{aa_i}|}{|C|} \tag{1}$$

$$\bar{R}_C = \sum_{aa=0}^{R} \bar{R}_{aa,C} \tag{2}$$

5.2.7 Experimental expression and solubility study of synthetic AKR sequences

Of the 30 enzyme sequences in SynthAKR30, 25 were successfully cloned and transformed into a host organism to assess how well they express. Specifically, these 25 sequences were assembled from codon optimised gene fragments provided by Twist Biosciences into pET28a vectors via NdeI and XhoI restriction sites. Sequence verified plasmids were then transformed into *Escherichia coli* strain BL21(DE3) onto kanamycin containing agar plates.

A single colony from each of the 25 synthetic-sequence containing plates was picked and grown for 8 hours in 10mL LB media at 35°C, induced with isopropylthio-β-galactoside (IPTG) and left to continue growing for 16 hours at 25°C. 1.5mL of cell culture was harvested via centrifugation at 13,000 rpm for 1 minute using a SLA-3000 rotor, resuspended in 0.3mL pH 7.5 sodium phosphate buffer, and then finally lysed via sonication at 4°C. Total cell fraction sodium dodecyl sulphate–polyacrylamide gel electrophoresis (SDS-PAGE) samples - which contain all of the cell's contents - were prepared from 15μL of the sonicated material. The remainder was then centrifuged for 5 minutes at 13,000 rpm using a SLA-3000 rotor.

Finally, cell free extract samples were prepared from 15μL of the supernatant. Samples were boiled for 5 minutes in a water bath and run on 12% acrylamide SDS-PAGE gels. On the gels, the total cell fraction samples showed the level of protein expression, while the cell free extract columns showed the level of protein solubility.

5.3 Results

5.3.1 Reconstruction of native sequences

An initial objective of the discriminant autoencoder built in this work is to encode enzyme sequences into condensed bottleneck representations, and reconstruct (i.e. decode) them back to their original sequences at a high degree of accuracy. Through the training process, the autoencoder will learn important features that summarise proteins in the bottleneck. The reconstruction accuracy of the autoencoder on the training and testing data was assessed by aligning reconstructed sequences to their corresponding native sequence in a pairwise global alignment. The pairwise identity resulting from these alignments is therefore interpreted as the reconstruction accuracy.

The average reconstruction accuracy for the training and testing dataset can be seen in Table 5.3, along with the number of reconstructed sequences that have at least one gap in their respective alignment. Widespread gapped sequences would represent a type of error indicating the autoencoder could not reconstruct sequences back to their original length, due to insertions and deletions, and therefore could not learn protein length as a basic feature.

Table 5.3: Table showing the reconstruction performance of the discriminant autoencoder for the training and testing datasets.

Dataset	Size	Average Reconstruction Accuracy	Gapped sequences
Training	244,514	98.10±1.0	74
Testing	5,309	98.02±0.9	1

The sequence reconstruction for both training and testing datasets is almost exact, with average reconstruction accuracies above 98% for both. Overfitting on the training dataset is unlikely due to the high average reconstruction accuracy also shown for the testing dataset, which is made up of unseen data, as described in section 2.4. Also, the number of sequences with gaps is negligible for both datasets, with 99.97% and 99.99% of the training and testing sequences reconstructed exactly to the length of their native sequences, respectively.

These results therefore show that there is no loss of significant information when sequences are encoded into the bottleneck layer, as enzyme sequences are reconstructed to a high degree of accuracy when passed through the decoder. These results also imply that the main error type introduced in the decoding process is mismatches, with negligible insertions and deletions. These mismatches also do not occur in large enough numbers to put into question the autoencoder's ability to encode and decode sequences accurately.

To verify whether this high level of reconstruction is consistent across different enzyme lengths, the reconstruction mismatch rate was plotted against sequence length for both training and testing datasets as a scatterplot, which can be seen in Figure 5.8, with the red line representing the average mismatch rate $\bar{m}$. The mismatch rate is the complement of the reconstruction accuracy e.g. if a sequence is reconstructed at 98% accuracy, its mismatch rate is $1 - 0.98$ or a mismatch rate of 0.02. Indeed, there is no noticeable pattern of a worse mismatch rate at particular lengths, bar some very short sequences in the training dataset with lengths below 100 residues. One reason for these few exceptions could be due to a negligible amount of enzymes with such short sequence length in the training dataset.

A similar length-based assessment was performed on the discriminant, with the precision, recall, and F1-score being computed for the predictions made by the discriminant

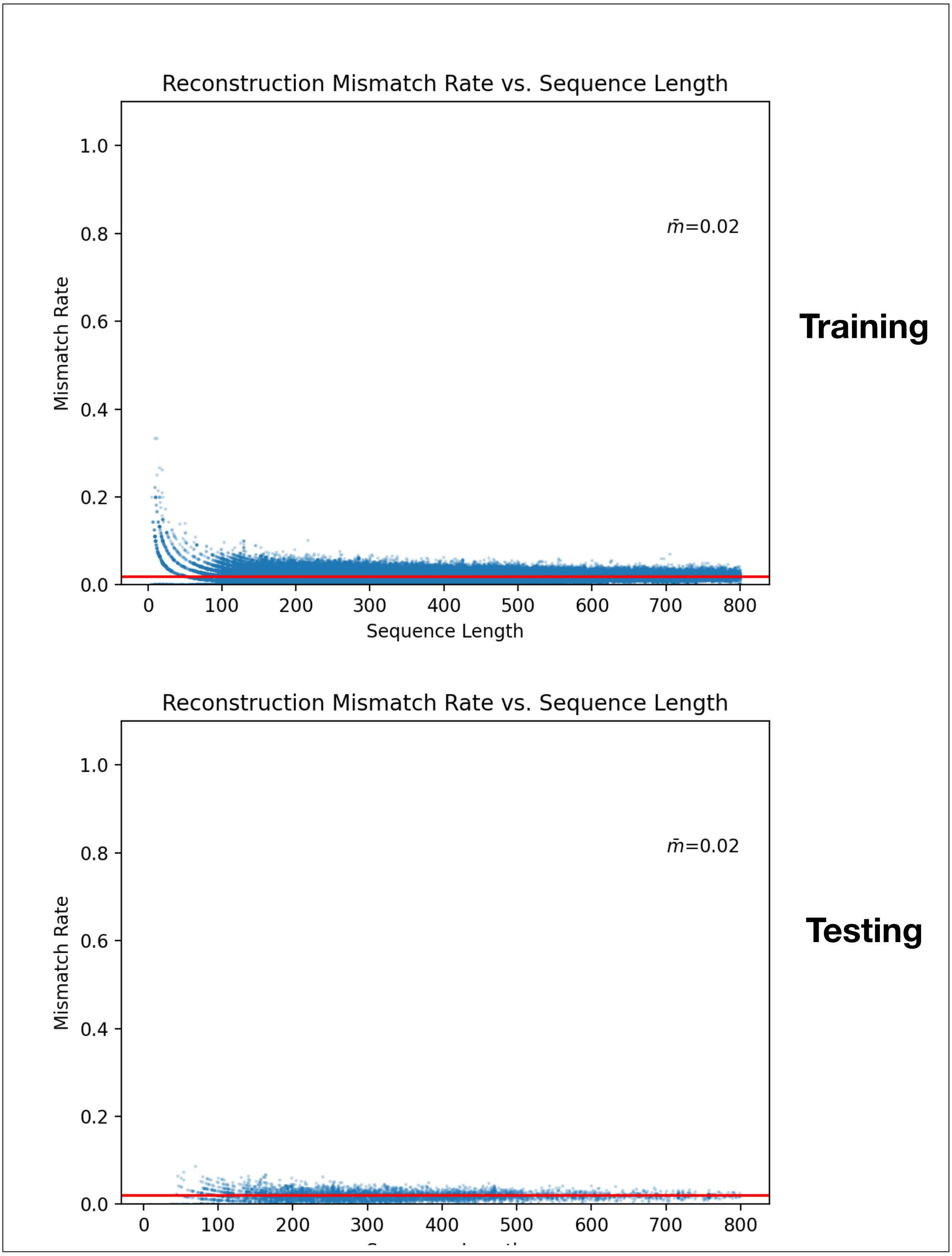


Figure 5.8: Scatterplots showing the mismatch rate versus sequence length for both the training and testing datasets. There is no significant correlation between length and mismatch rate except for some small sequences below 100 residues in the training dataset, with the average mismatch rate $\bar{m}$ being equal to 0.02 mismatches per sequence.

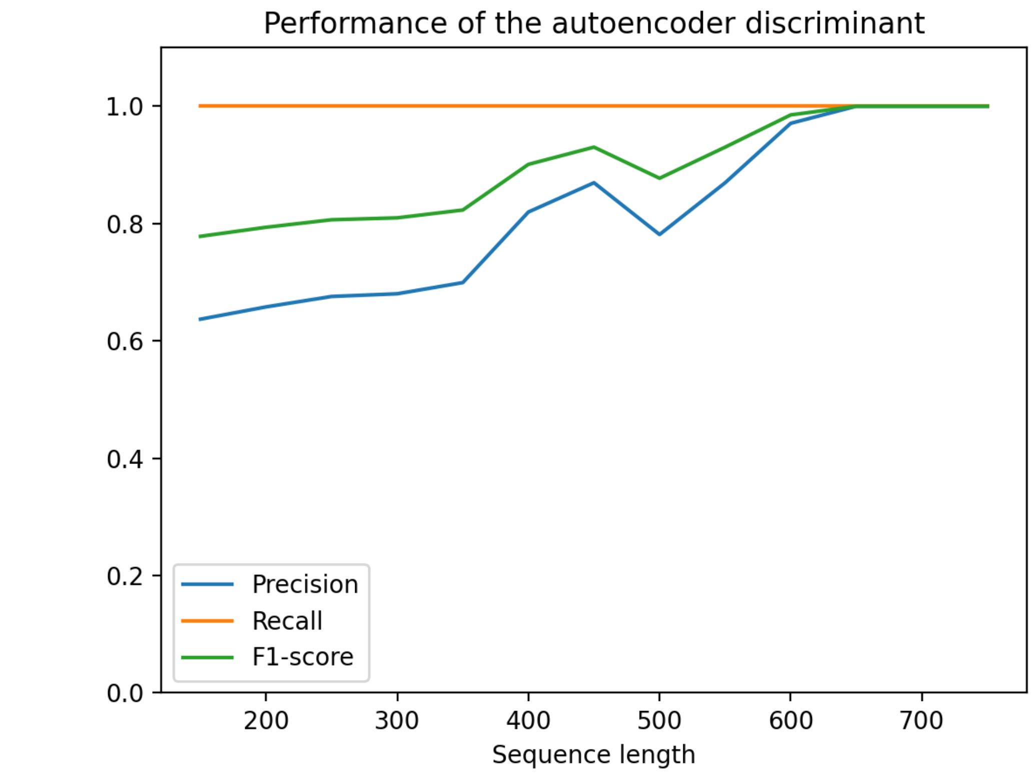


Figure 5.9: Plot showing the performance of the discriminant autoencoder based on sequence length using precision, recall, and F1-score. Native sequences from UniProt were sampled for different length ranges, and random sequences of similar sizes were generated. All these sequences were passed to the discriminant, and assessed based on the following definitions: true positives are when a native sequence is predicted as native, false positives are when a random sequence is predicted as native, true negatives are when a random sequence is predicted as random, and false negatives are when a random sequence is predicted as native. The recall is always flawless while the precision performs better for longer sequences.

for the sampled enzyme sequences (as described in section 5.2.6), plotted onto Figure 5.9. A noticeable imbalance between recall and precision can be seen, as the recall is equal to 1 across all sequence lengths, while the precision starts at 0.6 for the sequence length range 150-200. However, as the sequence length increases, the precision also increases, especially from 350 residues and above. The precision does decrease for the 450-500 range, before increasing again and converging with the recall from a sequence length of 600 and above.

A potential reason for this pattern lies enzyme sequences increase in the length distribution of the training dataset. As shown in Table 5.1, the average sequence length of the training dataset is of 351.5 residues. Therefore, the autoencoder is likely to better distinguish between native and random enzyme sequences of that length and above due to more experience. Also, the trend of higher precision at higher sequence lengths could be due to a higher amount of recognisable features the bigger an enzyme sequence is. Therefore, while the discriminant could perform better for shorter sequence lengths, this data could still imply the autoencoder has learned important features that enzymes should have.

To assess the similarity of the native and reconstructed sequence classes, two dimensionality-reduction techniques were used: PCA and t-SNE. These plots can be seen in Figure 5.10. If sequences are being reconstructed accurately, the two groups should not separate. Indeed, for both the PCA and t-SNE plots not only do the native and predicted sequences not separate into clear groups, there is high overlap between native and predicted sequences in where they are laid out in two-dimensional space. This result is further confirmation that the discriminant autoencoder learned how to accurately reconstruct protein sequences from the smaller condensed latent space.

5.3.2 Overview of the SynthAKR30 dataset

The SynthAKR30 dataset was sampled from the synthetic sequences after using the filtering pipeline described in section 5.2.5, with 15 sequences from Clade1 and 15 sequences from Clade2. The primary sequences of the SynthAKR30 enzymes can be seen in the Appendix section C.

In Figure 5.11, the distributions of three different scores can be seen for each clade: the one-match e-value, the one-match sequence identity, and the QMEAN of their generated SWISS-MODEL structures. These scores provide an overview of how similar to native enzyme sequences the SynthAKR30 are, based on sequence alignments for the e-value and pairwise sequence identity, and on homology modelling for the QMEAN.

In terms of e-value, the scores are high for both clades, with distributions ranging from around 1e^{-}300 to 1$e-375$ for Clade1, and from around 1e^{-}300 to 1e^{-}400 for Clade2. Clade1 also has two outliers, with one sequence with a one-match e-value of around 1e^{-}150 , and another of around 1e^{-}400. The distribution of e-values is more spread

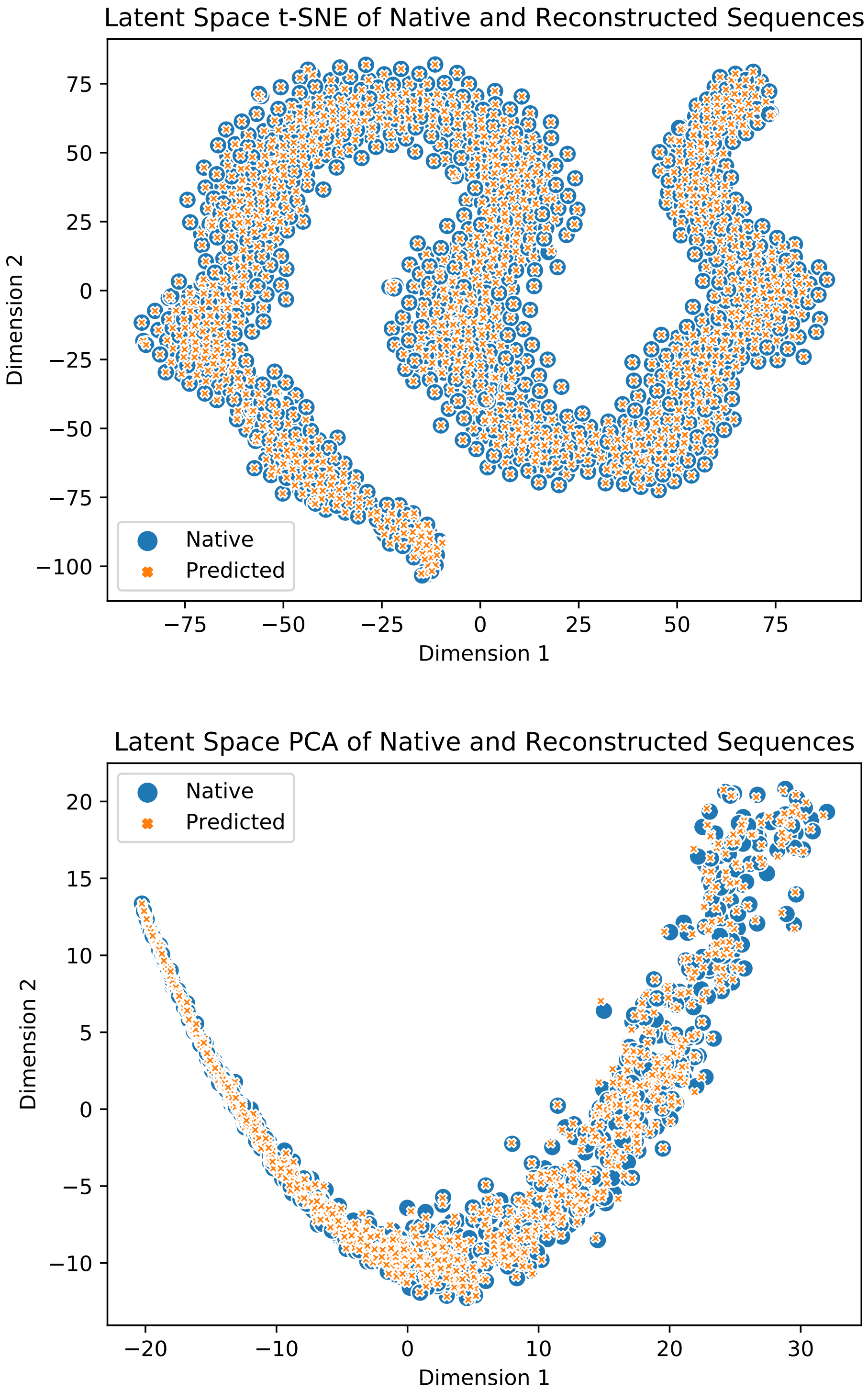


Figure 5.10: Dimensionality reduction plots for native and reconstructed sequences. Both the t-SNE and PCA plots confirm an essentially flawless overlap between native and reconstructed sequences, with many sequences and their reconstructed counterparts having extremely similar values.

out for the synthetic Clade2 sequences. In terms of the one-match sequence identity, Clade1's distribution leans slightly higher than Clade2's, with ranges from around 0.72 to 0.83 for Clade1 and 0.67 to 0.77 for Clade2. Clade1 yet again has two outliers, one at each end of the distribution. Overall, for both of these two similarity measures, the SynthAKR30 sequences have high sequence similarity to their respective one-match AKR59 enzymes.

Finally, the QMEAN score distribution is skewed higher for the Clade1 sequences than for Clade2, with a range from around -3.0 to around -1.5 for the former, and from -5 to -1.5 for the latter. Such a result implies that it is harder for SWISS-MODEL performing homology modelling for Clade2 sequences, likely due to a lack of structural templates to rely on. However, this result does not imply that Clade2 enzymes sequences are less likely to be functional, especially since both clades peak around the same QMEAN score, and since both clades also have highly scoring one-match e-value and sequence identity distributions.

5.3.3 Comparative analysis of SynthAKR30 and AKR59

Since the sequences of SynthAKR30 were generated using the two clades of AKR59, a comparative analysis to see how the predicted sets compare to their native sets could indicate how likely the synthetic sequences are to being viable enzymes.

First, SSNs were built for the union of AKR59 and SynthAKR30 at sequence identity thresholds of 40%, 50%, 60%, and 70% (Figure 5.12). There are zero edges between the native Clade1 and Clade2 sequences, an observation that the synthetic sequences also emulate across all thresholds, and therefore showing consistency between native and synthetic sequences. Also, it is clear from these SSNs that the distribution of the synthetic sequences does depend on the distribution of the template sequences - most of the synthetic sequences of each clade are part of the largest connected component for that clade (indicated by orange arrows).

The density of the subgraphs for native and synthetic sequences of each clade at each threshold can be seen in Table 5.4. When comparing the clades, it is clear that Clade1 sequences are more densely connected than Clade2 for both native and

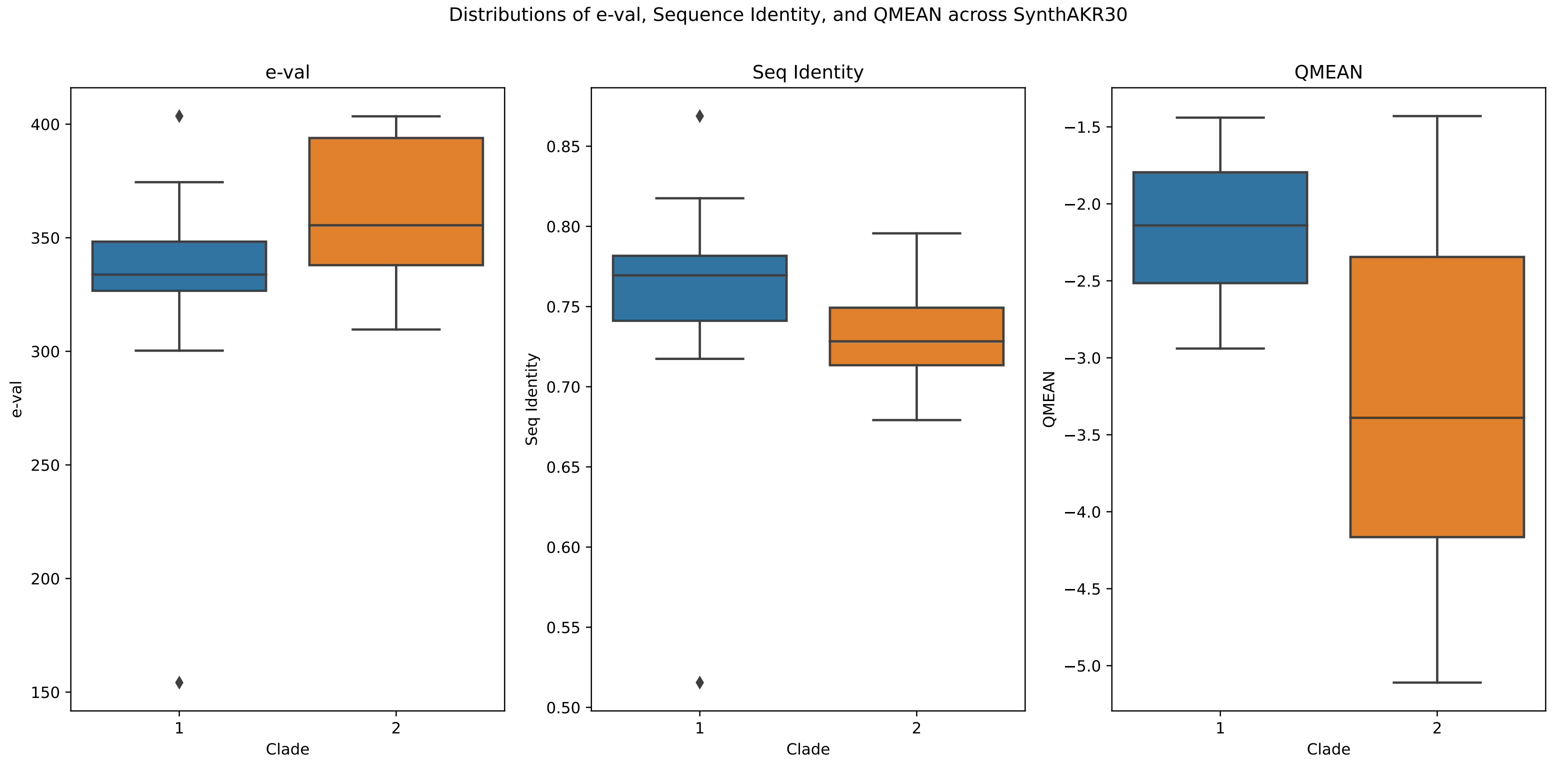


Figure 5.11: Boxplots of the one-match e-value, one-match sequence identity, and QMEAN distributions across the two clades making up the SynthAKR30 sequences. Both clades have ranges for the e-value and sequence identity that imply a high similarity to template sequences. It is also clear that Clade1 sequences have better available tertiary structure templates due to a narrower but higher score distribution of QMEAN scores.

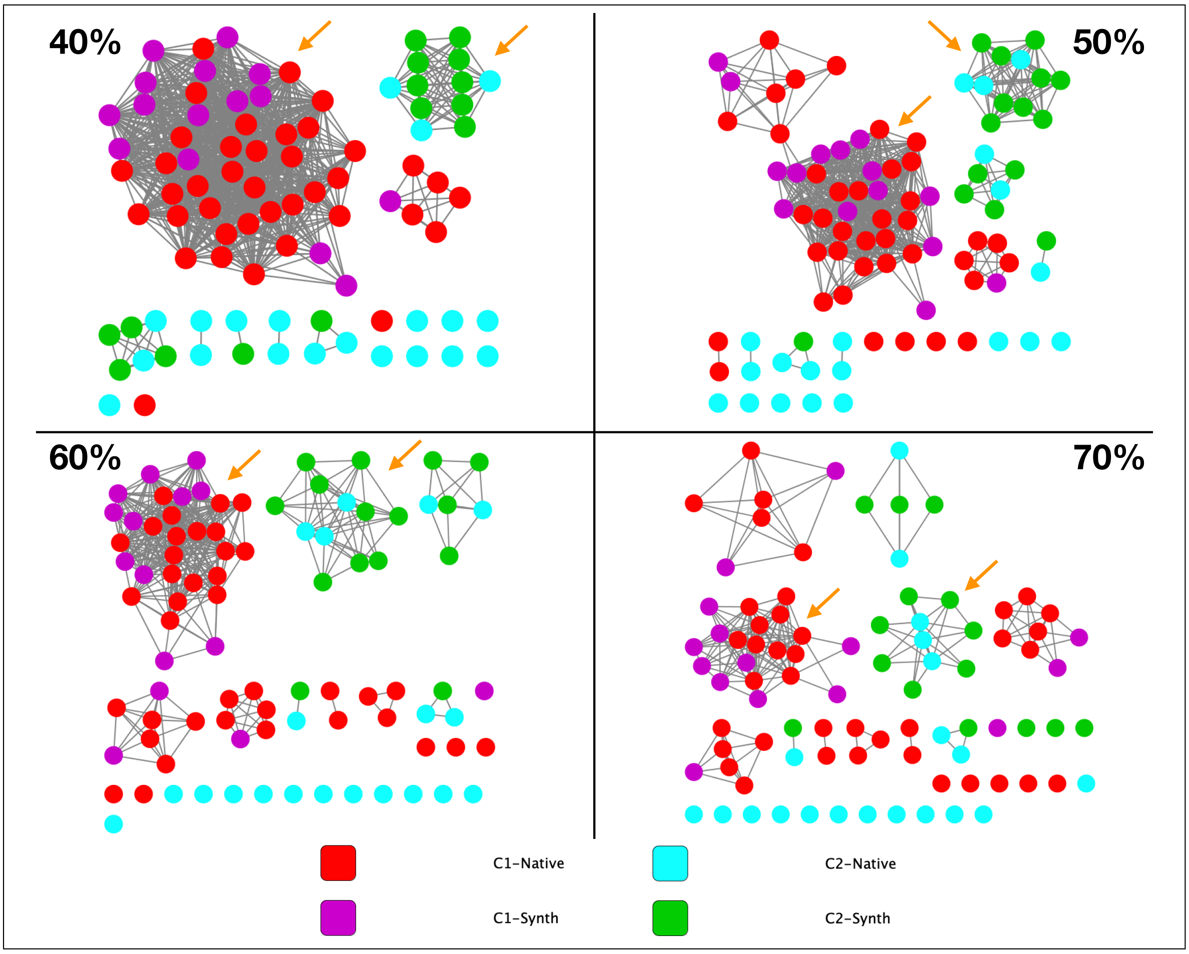


Figure 5.12: SSNs at 40%, 50%, 60%, and 70% identity thresholds for the AKR59 and SynthAKR30 sequences. There are no edges between the two clades for any threshold for the native sequences, an observation repeated by the synthetic sequences. Most synthetic sequences are part of the larger connected components of each clade, implying that sequence bias will affect the diversity of generated sequences.

Table 5.4: Table containing the SSN subgraph density for both clades, for both AKR59 and SynthAKR30, and across identity thresholds of 40%, 50%, 60%, and 70%.

Threshold	**40%**		**50%**		**60%**		**70%**	
Clade	**Clade1**	**Clade2**	**Clade1**	**Clade2**	**Clade 1**	**Clade2**	**Clade1**	**Clade2**
AKR59 subgraph density	0.648	0.037	0.310	0.037	0.242	0.026	0.128	0.026
SynthAKR30 subgraph density	0.638	0.400	0.371	0.400	0.305	0.219	0.105	0.019

synthetic sequences except for threshold 50%, where the Clade2 synthetic subgraph is slightly more dense than its Clade1 counterpart. This pattern being present in AKR59 and also mostly present in SynthAKR30 reinforces the hypothesis that the synthetic enzyme sequences are similar to their respective templates.

Finally, for all thresholds, the synthetic Clade2 subgraphs have higher density than their respective native subgraphs. This observation is made evident by the relatively high amount of edgeless nodes in the C2-Native subgraph, and implies a less diverse representation of the Clade2 sequence space in SynthAKR30.

This less diverse representation of its template's sequence space compared to Clade1 is likely due to a higher baseline sequence diversity in the native Clade2 enzymes, as shown by its comparatively lower densities for the latter across all thresholds. As the latent space of a template set is sampled from to generate synthetic enzymes, sequence space that is better represented in the template is more likely to be sampled from, as evidenced by the dense components shown in Figure 5.12. An implication of the latter observation is that the latent space sampling method used here can be affected negatively by sequence bias. However, given that the synthetic Clade2 sequences are still less dense than synthetic Clade1 sequences across most thresholds, a higher amount of catalytic diversity would stilll be expected in the synthetic Clade2 enzymes.

The evolutionary relationships between synthetic sequences and their native templates was explored using a phylogenetic tree, which is shown in Figure 5.13. This tree still displays a two-clade structure after the addition of the SynthAKR30 sequences. Much like in the SSNs, synthetic sequences clustered with their respective templates unanimously.

Most of the synthetic sequences across both clades on this tree can be traced back to a common ancestor that is evolutionary close to at least one native sequence. This is evidenced by negligible branch lengths from native sequences to said common ancestors (highlighted in orange). This observation implies that the native sequences phylogenetically closest to synthetic sequences function similarly to direct ancestors. This observation could imply that the autoencoder is sampling from the space around individual template sequences, resulting in modified versions of them.

For further analysis on where the synthetic sequences lie relative to the native sequences, two dimensionality reduction plots were created out of the latent space of both AKR59 and SynthAKR30: t-SNE and PCA, which can be seen in Figure 5.14. The first observation of note is that on both plots there is a clear separation between the two clades for both native and synthetic sequences, with synthetic sequences yet again being grouped up with their respective template sequences. Also, as was discovered from the phylogenetic tree (Figure 5.13), synthetic enzymes are in close proximity

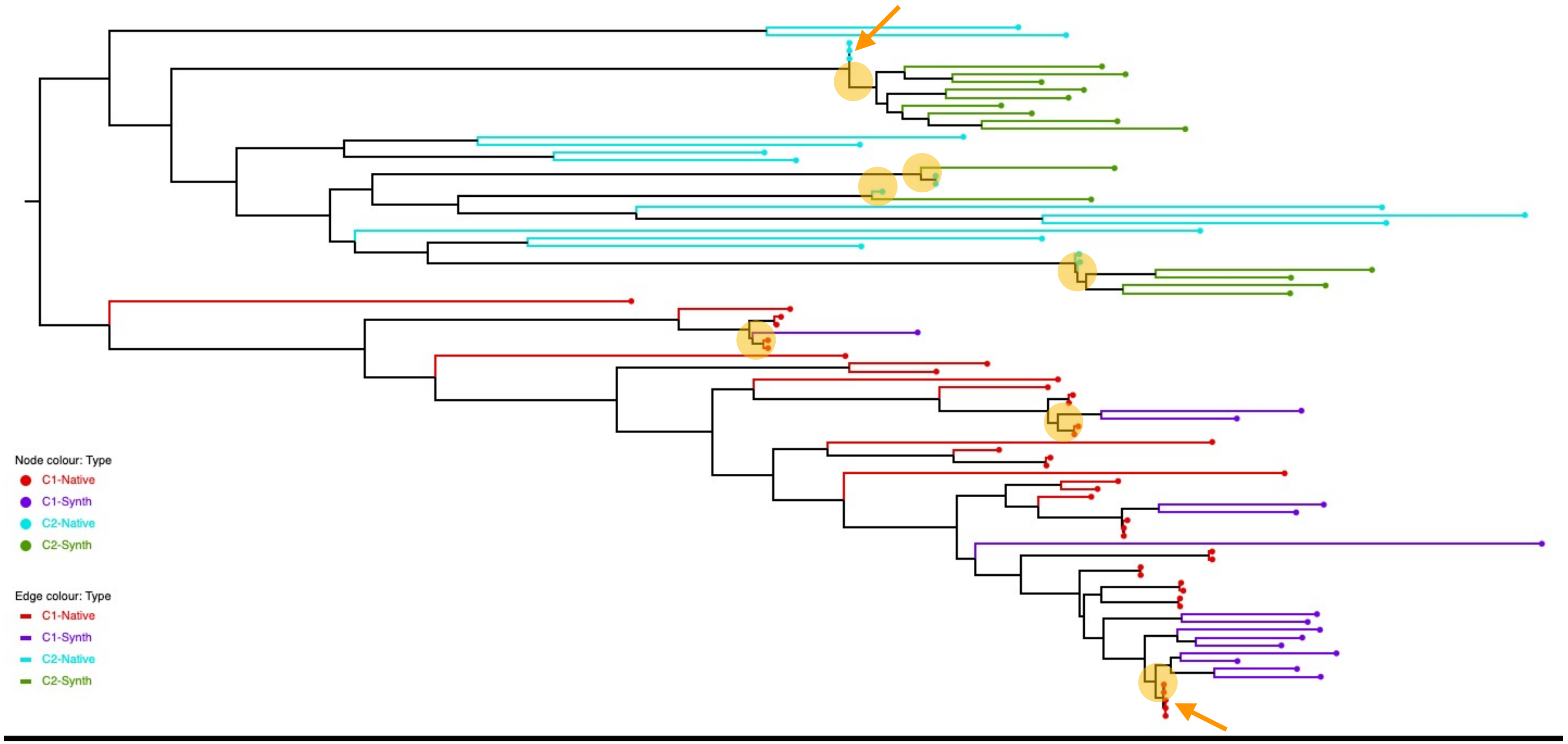


Figure 5.13: Phylogenetic tree for the AKR59 and SynthAKR30 sequences. The original two-clade structure seen in Figure 5.4 remains after adding the synthetic sequences, implying the synthetic sequences fit into this clade structure similarly to native ones. Also, similarly to the SSNs in Figure 5.12, synthetic sequences were clustered with their respective templates unanimously. Furthermore, as highlighted by the ancestor nodes in orange, most of the synthetic sequences can be traced back to an 'ancestor' that is very close to template sequences, implying these native sequences are almost direct ancestors. Such an observation provides clues that it is the latent space around individual sequences that gets sampled, generating synthetic sequences that have specific templates as a base.

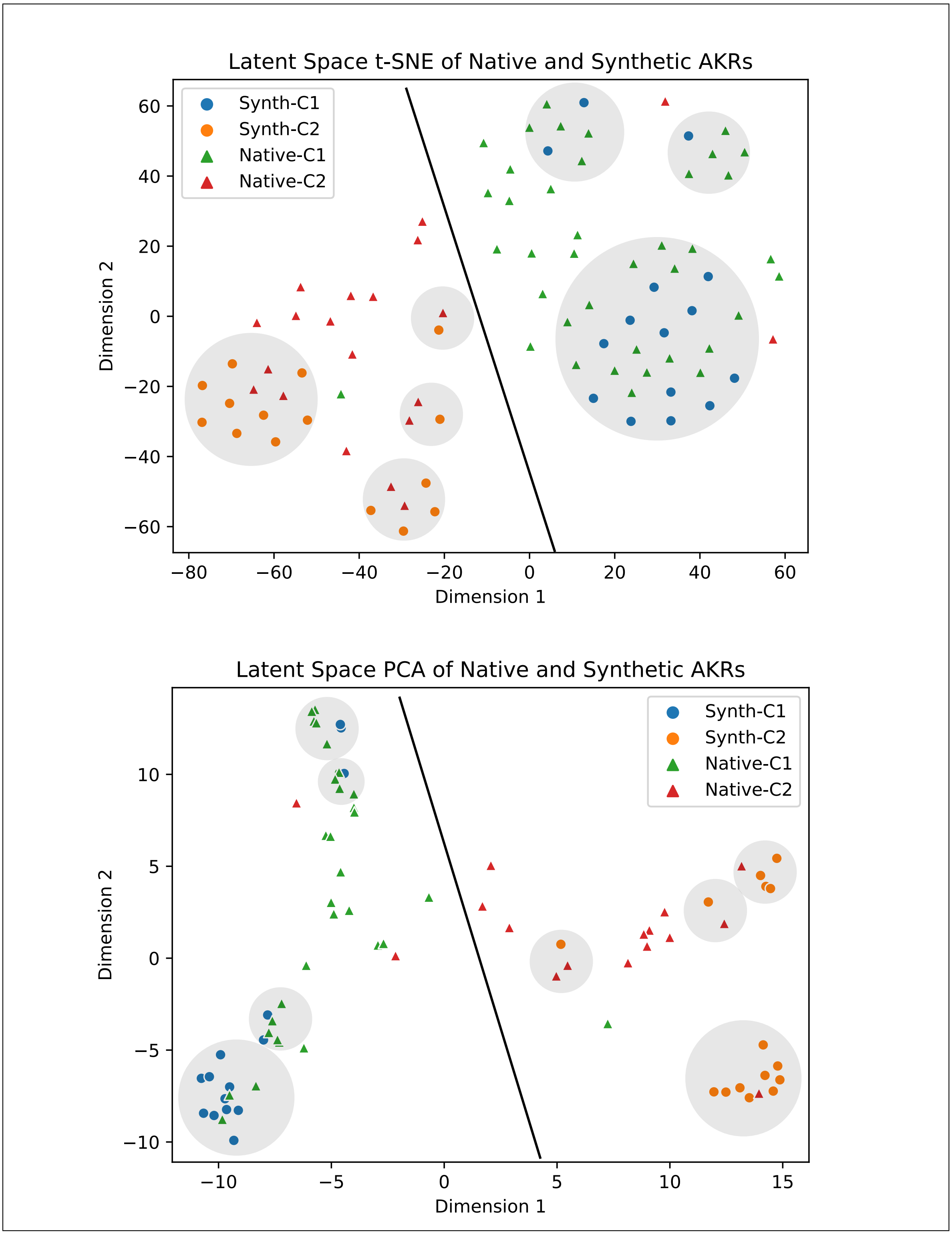


Figure 5.14: Dimensionality reduction plots for native and reconstructed sequences. There is a clear separation between the two clades for both native and synthetic sequences. Also, all synthetic sequences are in close proximity to at least one native sequence, giving further credence to the hypothesis that the sampling of generated sequences occurs with individual templates as a base.

to at least one native enzyme, instead of being spread out across the clade space randomly. This observation is consistent with the hypothesis that synthetic sequences are generated directly from the space around individual template enzymes.

In conclusion, it is clear that sequence similarity patterns existing in AKR59 - such as Clade1 being less diverse than Clade2 - also exist for SynthAKR30. The distribution of sequences that are viable across the templates was also discovered to depend highly on the sequence distribution of the templates themselves. Finally, the results of this section also point at a possible mechanism that the autoencoder uses to generate novel sequences, which is to directly pick around the latent space of individual template sequences. This mechanism clearly results in synthetic enzymes that are highly similar to the native template enzymes, with some modifications.

5.3.4 Functional and structural predictions for the SynthAKR30 sequences

Many computational tools exist for the sequence-based prediction of function. It may therefore be possible to verify the functional viability of the SynthAKR30 sequences *in silico* by using tools which predict function from primary sequence, examples of which were discussed in section 2.3 of the Background.

First, the synthetic sequences were analysed with the web-tool DEEPre [28], which is an Enzyme Commission (EC) classifier. The resulting predictions can be seen in Table 5.5. While DEEPre makes very confident predictions of the first three levels of an EC number (section 2.3), the fourth and most specific number is a more complicated problem. Nonetheless, for the purpose of this study, the predictions made by DEEPre are assumed to be true to make assess the possible catalytic diversity of the SynthAKR30 dataset. The set of predicted EC classes was also compared to the set of EC annotations that the AKR59 enzymes contain on Swiss-Prot. This comparison can be seen in Table 5.6.

Across both clades, the synthetic sequences were predicted as having seven different EC classes. Table 5.6 shows that all seven of the predicted functions are valid AKR functions, which we would expect for functionally viable sequences. As was predicted

Table 5.5: Table showing the DEEPre-predicted EC classes for the SynthAKR30 sequences. The EC classes in red represent the four predicted reactions that are unique to SynthAKR30 relative to AKR59.

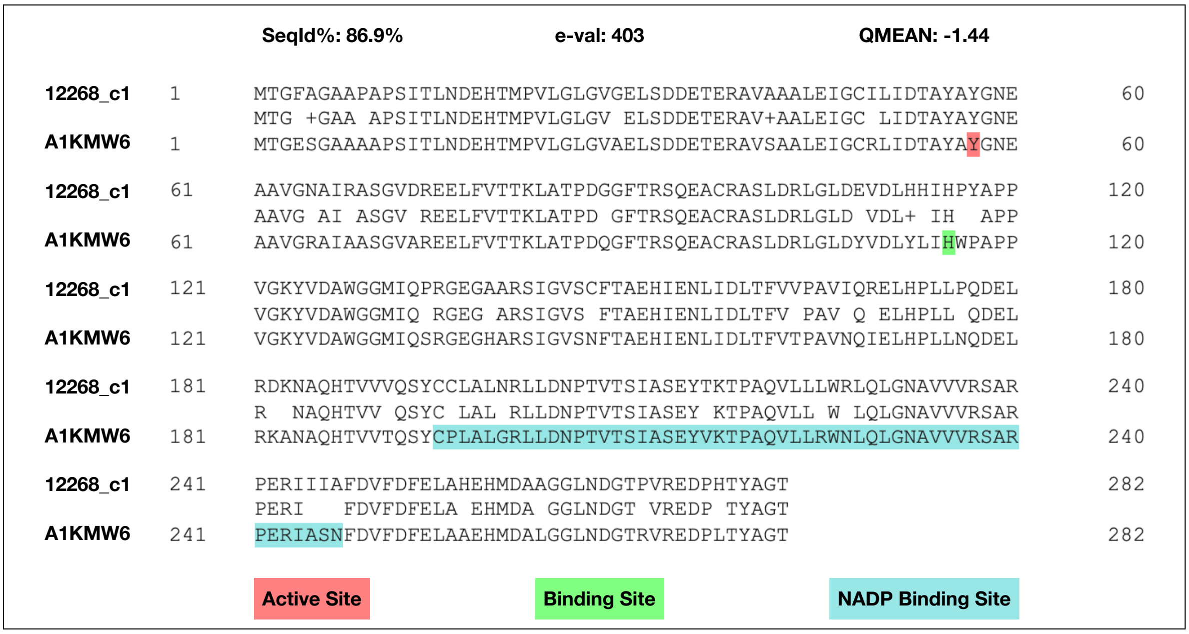
Synthetic Sequence	Predicted EC	Synthetic Sequence	Predicted EC
41284_c1	**1.1.1.188**	**4466_c2**	**1.1.1.2**
12268_c1	1.1.1.274	**9223_c2**	**1.1.1.316**
23919_c1	1.1.1.274	**26444_c2**	**1.1.1.317**
11553_c1	1.1.1.346	**2957_c2**	**1.1.1.317**
11719_c1	1.1.1.346	**33894_c2**	**1.1.1.317**
16119_c1	1.1.1.346	**45873_c2**	**1.1.1.317**
23716_c1	1.1.1.346	**49250_c2**	**1.1.1.317**
26539_c1	1.1.1.346	**9840_c2**	**1.1.1.317**
32249_c1	1.1.1.346	13192_c2	1.1.1.65
36404_c1	1.1.1.346	23504_c2	1.1.1.65
42186_c1	1.1.1.346	23827_c2	1.1.1.65
45878_c1	1.1.1.346	26535_c2	1.1.1.65
6626_c1	1.1.1.346	26820_c2	1.1.1.65
8092_c1	1.1.1.346	43465_c2	1.1.1.65
9261_c1	1.1.1.346	5050_c2	1.1.1.65

Table 5.6: Table showing the comparative set membership of EC classes for AKR59 and SynthAKR30. The EC classes predicted by DEEPre for the SynthAKR30 are those shown in table 5.5, whereas the annotations of the AKR59 enzymes originate from Swiss-Prot. While there is an overlap between the EC classes represented by both sets, both datasets represent EC classes that are unique to themselves. All four of the EC classes unique to the synthetic sequences do represent aldo/keto reductase functions.

Membership	EC class	Description
Unique to AKR59	1.1.1.184	Carbonyl reductase (NADPH)
	1.1.1.122	D-threo-aldose 1-dehydrogenase
	1.1.1.218	Morphine 6-dehydrogenase
	1.1.1.107	Pyridoxal 4-dehydrogenase
	1.1.1.283	Methylglyoxal reductase (NADPH)
Unique to SynthAKR30	1.1.1.2	Alcohol dehydrogenase (NADP(+))
	1.1.1.316	L-galactose 1-dehydrogenase
	1.1.1.317	Perakine reductase
	1.1.1.188	Prostaglandin 11-ketoreductase
Present in both	1.1.1.274	2,5-didehydrogluconate reductase (2-dehydro-D-gluconate-forming)
	1.1.1.346	2,5-didehydrogluconate reductase (2-dehydro-L-gluconate-forming)
	1.1.1.65	Pyridoxal reductase

in section 5.3.3, the Clade2 synthetic sequences have more diversity in predicted EC classes, with four of the seven EC classes being unique to them (Table 5.5).

The AKR59 sequences have eight different curated EC classes on Swiss-Prot. Of those eight, almost half of them are also present in the EC classes predicted for SynthAKR30,



```
                     SeqId%: 86.9%                   e-val: 403                    QMEAN: -1.44

12268_c1   1     MTGFAGAAPAPSITLNDEHTMPVLGLGVGELSDDETERAVAAALEIGCILIDTAYAYGNE    60
                 MTG +GAA APSITLNDEHTMPVLGLGV ELSDDETERAV+AALEIGC LIDTAYAYGNE
A1KMW6     1     MTGESGAAAAPSITLNDEHTMPVLGLGVAELSDDETERAVSAALEIGCRLIDTAYAYGNE    60

12268_c1   61    AAVGNAIRASGVDREELFVTTKLATPDGGFTRSQEACRASLDRLGLDEVDLHHIHPYAPP   120
                 AAVG AI ASGV REELFVTTKLATPD GFTRSQEACRASLDRLGLD VDL+ IH  APP
A1KMW6     61    AAVGRAIAASGVAREELFVTTKLATPDQGFTRSQEACRASLDRLGLDYVDLYLIHWPAPP   120

12268_c1   121   VGKYVDAWGGMIQPRGEGAARSIGVSCFTAEHIENLIDLTFVVPAVIQRELHPLLPQDEL   180
                 VGKYVDAWGGMIQ RGEG ARSIGVS FTAEHIENLIDLTFV PAV Q ELHPLL QDEL
A1KMW6     121   VGKYVDAWGGMIQSRGEGHARSIGVSNFTAEHIENLIDLTFVTPAVNQIELHPLLNQDEL   180

12268_c1   181   RDKNAQHTVVVQSYCCLALNRLLDNPTVTSIASEYTKTPAQVLLLWRLQLGNAVVVRSAR   240
                 R  NAQHTVV QSYC LAL RLLDNPTVTSIASEY KTPAQVLL W LQLGNAVVVRSAR
A1KMW6     181   RKANAQHTVVTQSYCPLALGRLLDNPTVTSIASEYVKTPAQVLLRWNLQLGNAVVVRSAR   240

12268_c1   241   PERIIIAFDVFDFELAHEHMDAAGGLNDGTPVREDPHTYAGT                     282
                 PERI   FDVFDFELA EHMDA GGLNDGT VREDP TYAGT
A1KMW6     241   PERIASNFDVFDFELAAEHMDALGGLNDGTRVREDPLTYAGT                     282

                 Active Site                 Binding Site                NADP Binding Site
```

Figure 5.15: Pairwise alignment of the template ezyme A1KMW6 and the synthetic sequence 12268_c1. This synthetic sequence has the highest similarity to a template enzyme, with a sequence identity of 86.9%, a one-match -log(eval) of 403, and a QMEAN score of -1.44 The active site, binding site, and NADP binding site, as annotated by UniProt, is conserved for 12268_c1.

with five of them being unique to the native sequences (Table 5.6). There is therefore some overlap between the predicted EC classes of SynthAKR30 and the template sequences used to generate them.

Interestingly, the synthetic sequences also have EC classes that are unique compared to the native AKR30 set (Table 5.6). Furthermore, all four of these EC classes do also represent enzymatic functions that are considered part of the AKR family. The canonical AKR annotation is by no means complete, and the predicted EC classes are not necessarily true until experimentally confirmed. However, these four EC classes indicate that the autoencoder generated synthetic enzyme that are predicted to perform relevant catalytic functions that are not known to exist in the template sets.

Next, the meta-server I-TASSER [82, 187, 188] was used to perform an *in silico* analysis of synthetic sequence and structure. I-TASSER performs multiple different analyses, including tertiary structure modelling, active site detection, and substrate binding prediction, and is known to be a state-of-the-art method in these fields.

This analysis was limited to just one synthetic enzyme sequence due to I-TASSER being a computationally time-consuming platform. To maximise the chances of useful predictions from I-TASSER, the synthetic sequence most similar to a native template sequence was chosen. Consequently, 12268_c1, which is a Clade1 sequence most similar to the native sequence with the UniProt accession A1KMW6, a D-threo-aldose 1-dehydrogenase was selected. A pairwise alignment of these enzymes can be seen in Figure 5.15, with the active site, binding site, and nicotinamide adenine dinucleotide phosphate (NADP) binding site highlighted. 12268_c1 has a sequence identity of 86.9% with A1KMW6, a one-match e-value of 403, and its SWISSMODEL tertiary structure has a QMEAN score of -1.44, all of which are high quality scores, making it a suitable candidate for this study.

Key I-TASSER results for 12268_c1 can be seen in Figure 5.16. The normalized B-factor, which is a measure of residue thermal mobility and stability, is plotted for the whole sequence (Figure 5.16-A). Values of the B-factor below 0 (the red dotted line) are considered more stable, and the opposite when above 0. The start and ends of the sequence are seemingly unstable, as is expected [189], with most of the other residues having stable values under 0. This result indicates that 12268_c1 has a potentially highly stable tertiary structure.

One of the key results that I-TASSER displays is the top five tertiary structure models predicted. However, it is possible for this top five to converge to a single structure if the model is good enough. The confidence score (C-score) given by I-TASSER can range from -5 to 2, with higher scores signifying higher confidence in the predicted structure. Indeed, the predicted structure for 12268_c1 did converge to one structure, which can be seen in Figure 5.16-B. This converged model has a C-score of 1.43, which is relatively high, and therefore implies that 12268_c1 is likely to have a structure similar to one modelled by I-TASSER. This result implies that 12268_c1 is considered to be similar enough to a native AKRs that a structure can be confidently modelled for it by I-TASSER.

Part of the I-TASSER pipeline is to identify template experimental structures to help with predictions, using TM-align [80] to assess tertiary structure alignment quality, a tool which was described in section 2.3.3. The top template structure chosen by I-TASSER has the PDB ID 4OTK, and an alignment of its structure with the one predicted for 12268_c1 can be seen in Figure 5.16-C. The superposition of these two struc-

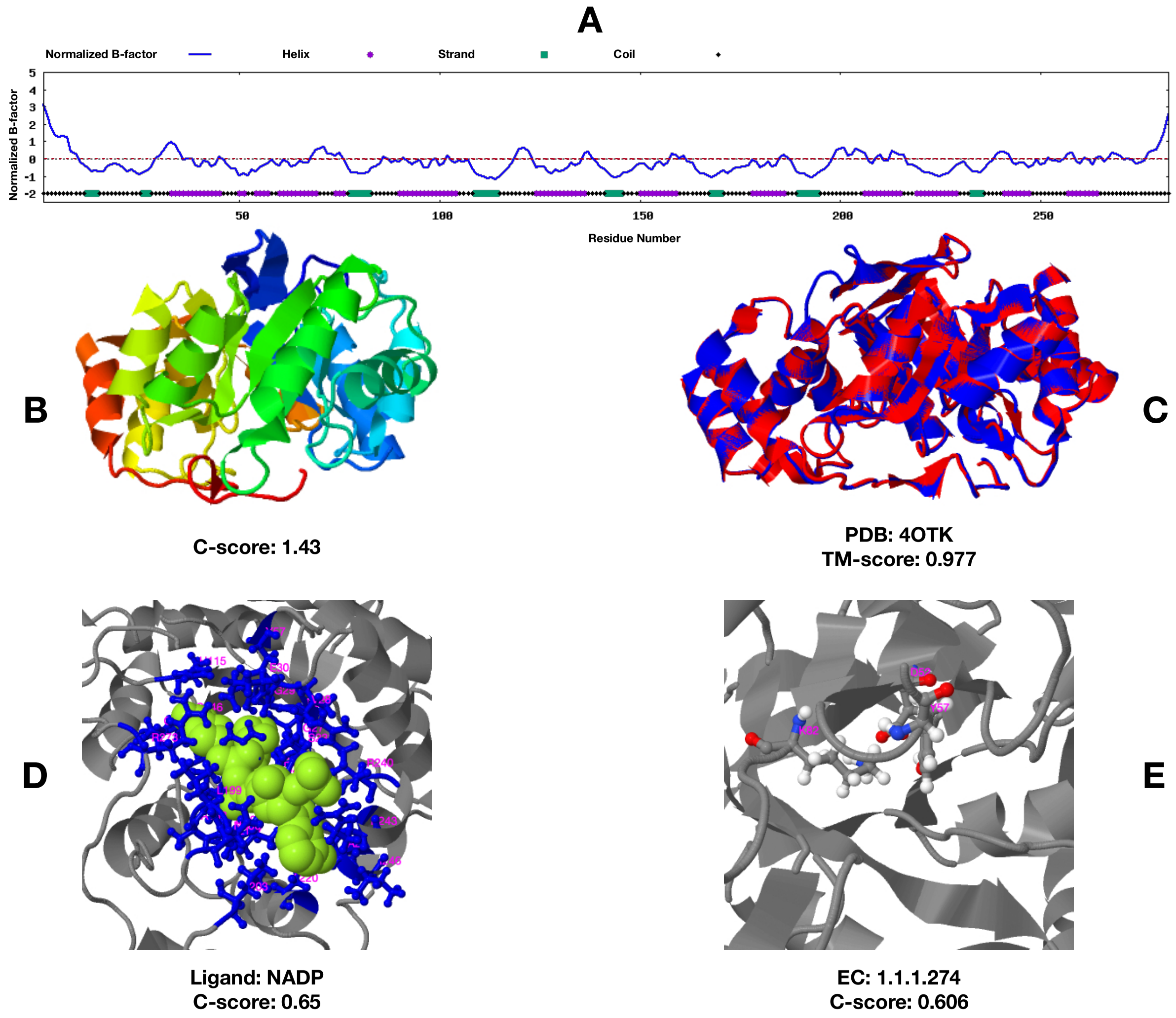


Figure 5.16: I-TASSER results for the 12268_c1 synthetic sequence. A- The normalized B-factor prediction of the sequence. The values of most residues being below 0 implies likely stable tertiary structure. B- Predicted tertiary structure of 12268_c1. I-TASSER's modelling converged to this structure at a C-score of 1.43, which is a high quality prediction. C- Superposition of the predicted structure of 12268_c1 and the best identified template, with the PDB ID 4OTK. With a TM-score of 0.977, the predicted structure is very highly similar to a native enzyme. D- Binding site prediction for the synthetic sequence. Specifically, NADP, which is the common cofactor in use by aldo-keto reductases, was correctly predicted with a C-score of 0.65. E- Active site prediction for the synthetic sequence. The EC-class 1.1.1.274 was predicted with a C-score of 0.606. Interestingly, this is the same class predicted by DEEPre (Table 5.6), showing agreement between two prediction tools about the potential function of this synthetic sequence.

tures is almost identical, and is made quantitatively evident through a high TM-score of 0.977. This score is further indication of not just a higher likelihood of structural viability for the synthetic sequence, but also of its viability as a potential AKR.

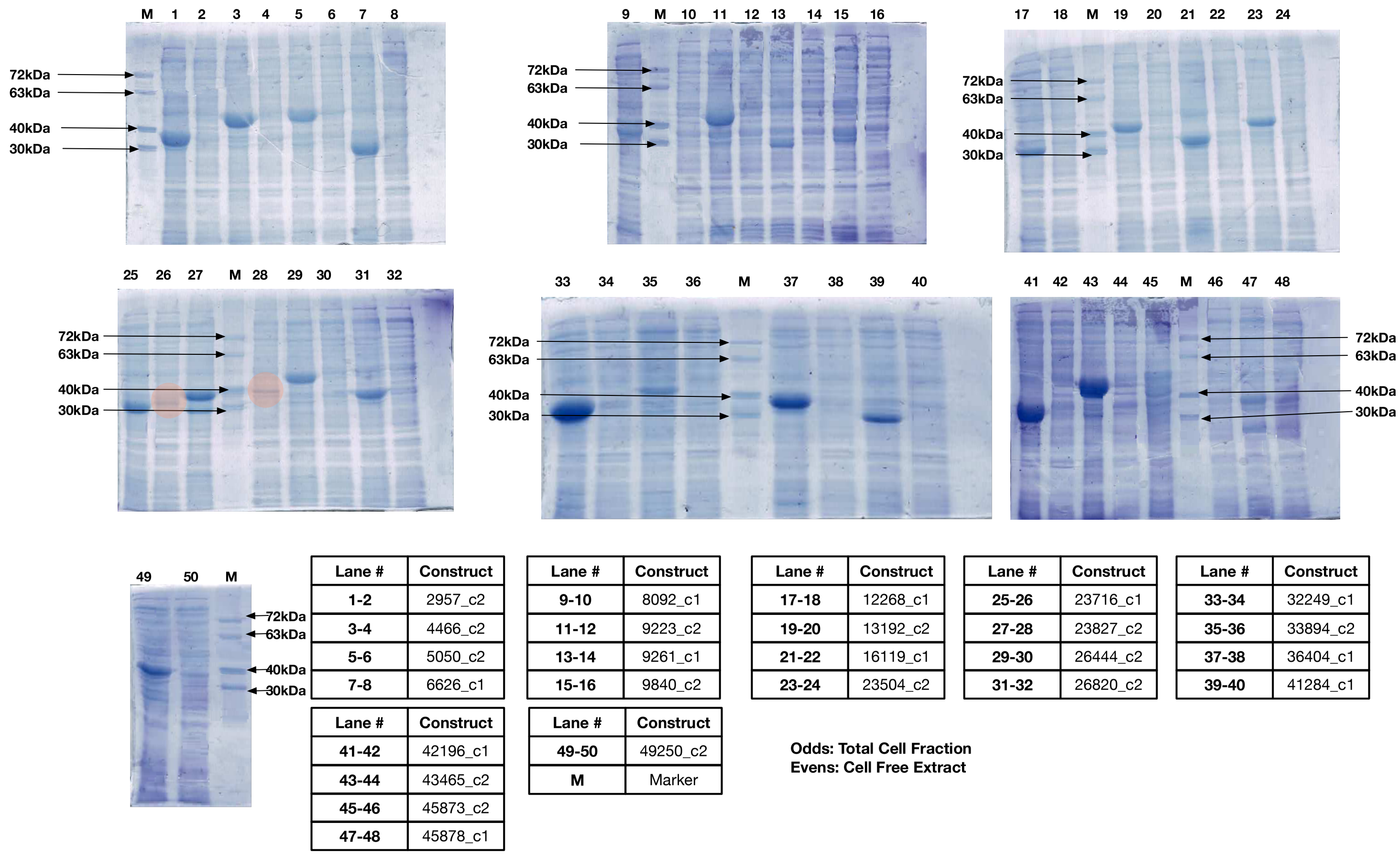
Finally, I-TASSER attempts functional predictions for input sequences by identifying binding and active sites, and potential molecules that could fit said sites, giving C-scores that range from 0 to 1, the higher the score the more confident the prediction. In Figure 5.16-D, the top binding site prediction can be seen as NADP, with a C-score of 0.65. AKRs always have either a nicotinamide adenine dinucleotide (NAD) or NADP binding site, and the top template of 12268_c1, A1KMW6, indeed bears the latter. Also, in Figure 5.16-E, the active site was predicted as bearing the EC class 1.1.1.274 with a C-score of 0.606, which is the same class predicted by DEEPre (Table 5.5), and is indeed an AKR function. This second prediction of the same potential function gives further credence to 12268_c1 being not just an AKR, but one that catalyses the reduction of 2,5-didehydrogluconate. Such assertions of function from different sources increases the confidence that SynthAKR30 enzymes could be functionally viable.

5.3.5 Experimental expression and solubility of SynthAKR30 proteins

A common method of analysing synthetic proteins is to overexpress them in some expression host system and then observe the level of expression using SDS-PAGE gels. 25 out of the 30 sequences from SynthAKR30 were successfully transformed into *E. coli* strain BL21(DE3). Their the level of expression was inspected using SDS-PAGE gels. Both the total cell fraction (TC) and cell free extract (CFE) were analysed, the latter of which helps indicate the level of solubility of individual proteins. The primary sequence and molecular weight of the different enzymes can be seen in the Appendix section C

The gels can be seen in Figure 5.17. Odd numbered lanes contain the TC samples, followed by even numbers being CFE samples e.g. lane 3 is the TC sample of 4466_c2 while lane 4 is its CFE sample. The marker lanes help identify the bands that correspond to the individual proteins based on protein weight.

The TC lanes of the gels show that almost all of the 25 synthetic enzyme proteins


M 1 2 3 4 5 6 7 8
9 M 10 11 12 13 14 15 16
17 18 M 19 20 21 22 23 24
25 26 27 M 28 29 30 31 32
33 34 35 36 M 37 38 39 40
41 42 43 44 45 M 46 47 48
49 50 M
72kDa
63kDa
40kDa
30kDa

Lane #	Construct
1-2	2957_c2
3-4	4466_c2
5-6	5050_c2
7-8	6626_c1

Lane #	Construct
9-10	8092_c1
11-12	9223_c2
13-14	9261_c1
15-16	9840_c2

Lane #	Construct
17-18	12268_c1
19-20	13192_c2
21-22	16119_c1
23-24	23504_c2

Lane #	Construct
25-26	23716_c1
27-28	23827_c2
29-30	26444_c2
31-32	26820_c2

Lane #	Construct
33-34	32249_c1
35-36	33894_c2
37-38	36404_c1
39-40	41284_c1

Lane #	Construct
41-42	42196_c1
43-44	43465_c2
45-46	45873_c2
47-48	45878_c1

Lane #	Construct
49-50	49250_c2
M	Marker

Odds: Total Cell Fraction
Evens: Cell Free Extract

Figure 5.17: SDS-PAGE gels of the 25 cloned synthetic sequences. The expression level in total cell fraction (TC) and cell free extract (CFE) was analysed, with odd numbered lanes showing the former, and even numbered lanes the latter. A four-step ladder with markers at 30kDa, 40kDa, 63kDa, and 72kDa was used in the lanes marked by M. After overexpression through induction by IPTG, it can be seen from the TC lanes that most of the synthetic sequences are able to overexpress. However, the CFE lanes comparatively have little to no expression, with light bands present only for 23716_c1 (Lane 26) and 23827_c2 (Lane 28). The implication from this result is that the synthetic sequences, while overexpressed, are not soluble, likely due to misfolding.

were able to overexpress, including some with very dense bands like 33249_c1 and 42196_c1. It can however also be seen that most of the proteins, whilst overexpressing, are not soluble, with far lighter CFE bands only present for 23716_c1 and 23827_c2 (highlighted in orange), with the rest of the proteins not having noticeable bands in the CFE lanes at the correct molecular weight at all.

While insoluble enzymes do not necessarily imply non-functional enzymes, one reason for proteins to be insoluble is because they are misfolded. Given that almost all of the 25 synthetic proteins tested here appear to be insoluble, it might therefore be a pattern for most of the sequences generated using the methods described in this chapter.

While it is difficult to clarify the reasons for such high levels of insolubility in the synthetic sequences without further laboratory experiments, *in silico* methods could offer some clarity. Therefore, an analysis of low complexity regions (LCR) was performed on both the AKR59 and SynthAKR30 datasets, where patterns of simple perfect amino acid repeats were identified, as repeats can have an impact on protein solubility [186].

In Figure 5.18, bar plots for different repeat sizes, for all 20 amino acids, for both clades, and for both AKR59 and SynthAKR30, can be seen. It is clear that for every repeat size, the synthetic sequences have significantly higher rates of repeats. Quantitatively, for both clades and for all repeat sizes, the synthetic $\bar{R}_C$ is far higher than the native $\bar{R}_C$, ranging from 0.73 to 37.67 for the former, and from 0.1 to 17.5 for the latter. While both native and synthetic sequences have significant amounts of repeats of size 2, there are multiple amino acids for which the synthetic $\bar{R}_{aa}$ is higher for both clades, such as for glycine (GG), valine (VV), and leucine (LL). This difference in repeat frequencies is even true for perfect repeats as long as 4 residues for both clades, but especially so for k=3. The source of these unexpected patterns of repeats is likely as to arise from incorrect artefacts of the decoding process, where smaller repeat patterns common to the template sequences are mistakenly extended more than necessary.

Interestingly, multiple of the hydrophobic residues like valine (V) and leucine (L) make up some of the highest amino acid repeats. Given that the more hydrophobic a compound is the more insoluble it is, such a detail merited further study. In Figure 5.19, a histogram showing the counts of hydrophobic repeats for native and synthetic sequences can be seen. It is clear yet again that the distribution of the number of

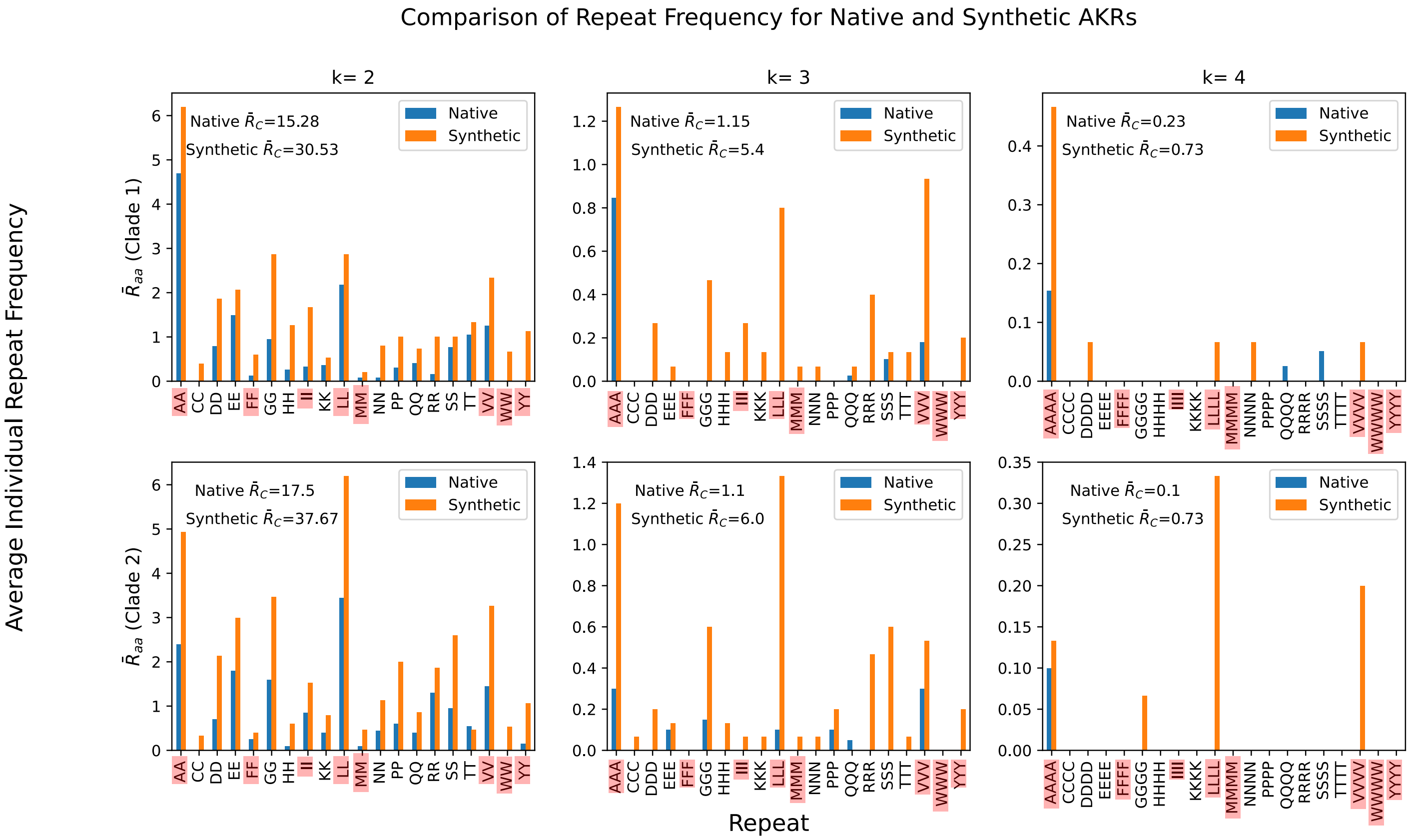


Figure 5.18: Bar plots comparing the average frequency of simple perfect repeats (SPR), $\bar{R}_{aa}$, in AKR59 and SynthAKR30 for the two different clades. Repeats highlighted in pink are of hydrophobic amino acids. The average number of repeats per sequence of a clade, $\bar{R}_C$, is also calculated for both datasets and clades. It can be seen that synthetic sequences have more SPRs than native sequences, across both clades and for all the different amino acids and repeat lengths. This observation is also true for hydrophobic repeats, which are likely to negatively affect the solubility of proteins even more.

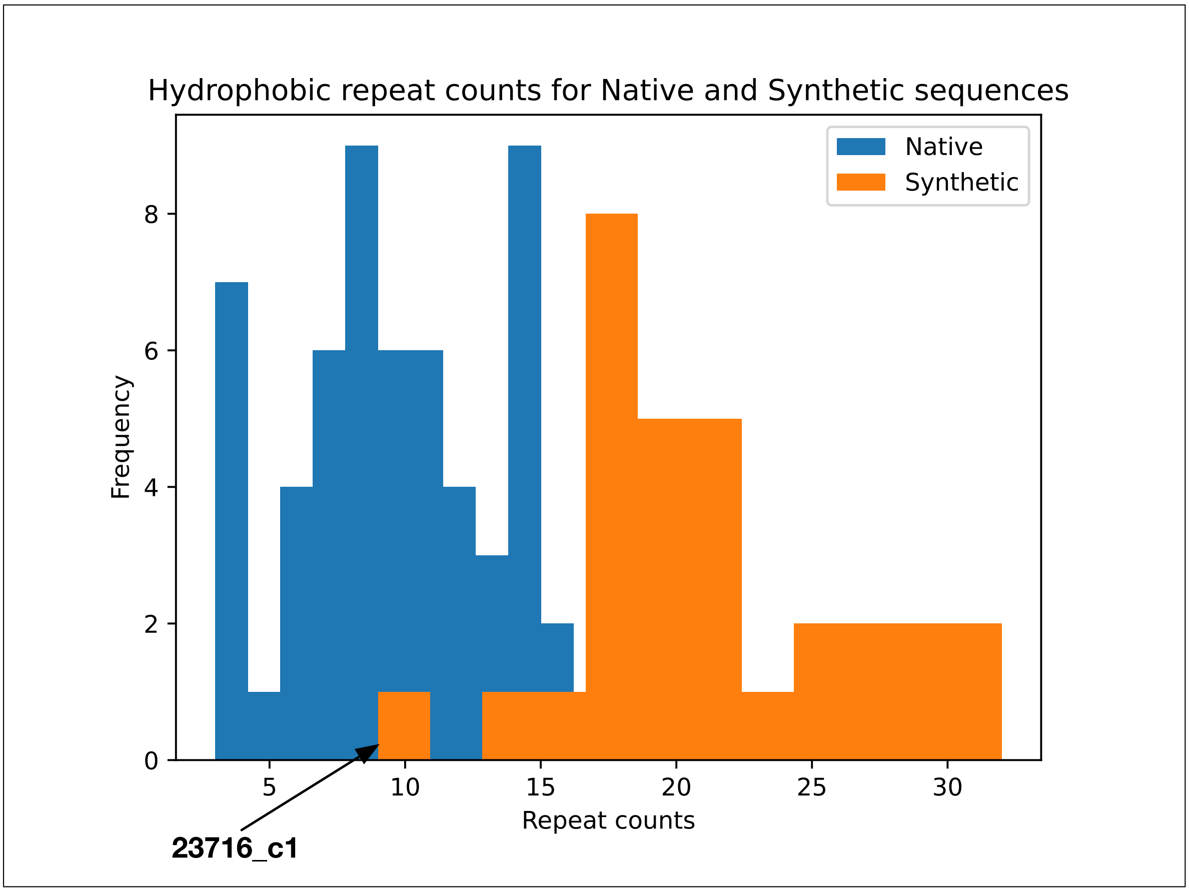


Figure 5.19: Histogram showing the distribution o hydrophobic repeats counts for the AKR59 and SynthAKR30 datasets. As was shown in Figure 5.18, synthetic sequences have a higher amount of repeats on average than native ones. However, there is one notable exception in the synthetic sequences, with 23717_c1 having just 9 different hydrophobic repeats, a number which is more in line with native sequences. This synthetic sequence is one of just two to show some amount of soluble expression, which could be evidence of such repeats being a principle reason for the lack of solubility in the synthetic sequences.

hydrophobic perfect repeats in the synthetic sequences are higher compared to the native sequences, with a mean of 21.13 hydrophobic repeats for the former and 9.71 for the latter.

There is however one exception: 23716_c1, which is indicated by an arrow, has just 9 hydrophobic repeats, which is in line with the native sequences. This enzyme in particular is one of the few synthetic sequences that displayed a noticeable level of solubility (Figure 5.17). This exception could imply that the synthetic sequences are insoluble partly due to a higher level of hydrophobic repeats than expected. Therefore, most synthetic enzymes were able to overexpress *in-vitro*, it is likely that identifiable characteristics of the synthetic sequences like the patterns of perfect repeats explain why most are insoluble.

5.4 Discussion and conclusions

While sequencing of environmental samples for the mining of potentially novel biocatalysts from metagenomes has become simpler over the last two decades, it is by no means an easy process, especially for particularly novel organisms and sequences. Studies generating functioning variants of specific enzymes [178] *in silico* have already been successful, but as far as the author is aware, have not been replicated on an enzyme family scale. Therefore, there is value in the synthetic generation of novel and viable sequences to artificially increase available panels of enzyme families.

To this end, this work consisted of a study into the viability of enzymes generated using a discriminant autoencoder neural network. Trained on all of the enzymes available on Swiss-Prot, the autoencoder was trained to reconstruct primary sequences after condensing them into smaller sets of features that are representative of proteins as a learnable concept, called a latent space, which it achieved with high accuracy. This latent space can then be sampled from to generate novel 'variants' of sequences from an input template set.

5.4.1 Strengths and limitations of the autoencoder approach

The output of the sampling of the latent space, after going through a strict filtering pipeline, is a set of automatically generated synthetic sequences that was shown to represent the sequence space around the template set. Specifically, high levels of similarity was observed between the native template aldo-keto-reductases used in this study and their synthetically generated counterparts, including similarity in sequence, predicted tertiary structure, and taxonomy. This similarity is a strength of the approach, as generating synthetic enzymes similar to a given template set is a requirement for potentially introducing novel properties to an enzyme family using this method.

Functional and structural predictions were performed on the sequences using state-of-the-art tooling like DEEPre [88] and I-TASSER [82], showing that all of the synthetic enzymes were predicted to belong to the AKR family, with features consistent with this designation like conserved catalytic residues and nucleotide-binding sites, predicted fold, etc., being present. Also, some of the more specific predictions like EC classes were confirmed by multiple sources, like in the case of the synthetic enzyme 12268_c1, showing that the approach used in this work is capable of generating enzymes similar enough in both hypothetical function and structure in the eyes of modern bioinformatics tooling.

To gain further experimental insight on their validity, the synthetic sequences were expressed in *E. coli*, and their levels of expression and solubility were observed. Most of the sequences were capable of overexpressing *in-vitro*, which is another noteworthy strength.

However, the majority of the synthetic enzymes were found to be insoluble, which is a significant limitation, as a likely implication of such widespread insolubility is that they are misfolded. At such low levels of solubility, it is complex to extract enough enzyme to perform characterisation studies. Therefore, it was not possible to test the activity profile of the synthetic enzymes, which is another limitation of this study. However, one likely reason for this weakness is a pattern of deleterious amino acid repeats that are seemingly inherent to the machine learning approach used to generate the sequences chosen for this work. An artefact of the training process such as this is one that can likely be refined away through iterating on the model, but also on improving the filtering pipeline in use.

Ultimately however, artefacts such as this are a likely indication that the neural network architecture is not complex enough to learn a problem as complex as enzyme structure and function. While the dimensions of the bottleneck layer relative to the original input have been significantly reduced, it is likely that a vector size of 200x32 is still too large. Indeed, a latent space of size 200x32 is still a vector of size 6400 when flattened - more than enough for a neural network to learn how to 'copy and paste' any native enzyme sequence, and therefore overfit. An indication of this happening is the significantly high average reconstruction accuracy displayed in Table 5.3.

In terms of methodology, while the training dataset is made up of over 200,000 manually curated sequences from Swiss-Prot, some sequences are very short. These short sequences potentially could have impacted the learning process negatively, as sequences of such length are likely to be fragments rather than full sequences. Also, the protocol for the laboratory experiments on the SynthAKR30 enzymes discussed in section 5.3.5 miss two crucial components: positive and negative controls. An enzyme known to function as an AKR, and known to be soluble under the laboratory conditions used in this work, would be an ideal positive control. Also, a soluble but non-AKR protein like Green Fluorescent Protein (GFP) would function well as a negative control.

Finally, despite a large amount of manual work that was spent parametrising and optimising the neural networks model, more systematic methods are available with enough compute power. For example, genetic algorithms have been used to optimise neural network parametrisation [190]. The simple model used here could be improved by this kind of system.

5.4.2 A *de novo* approach to artificially maximise the diversity of enzyme family panels

Although it was unfortunately not possible to get a definitive conclusion on the functional viability of the synthetic AKRs generated in this work, the results are promising overall. The synthetic enzymes were all recognised as being close enough to native AKRs by bioinformatics tools and that almost all could be overexpressed *in vitro*, implying that refinement of the training process to reduce regressive sequence artefacts could be a final hurdle for generating functional synthetic enzymes of a particular family. Such a possibility is backed up by the literature, with similarly produced synthetic sequences proven to add value to training sets [177] and functional synthetic malate dehydrogenases having very recently been constructed [178].

Therefore, a *de novo* approach for increasing panel diversity artificially using synthetic sequences created by an autoencoder could be formulated. Not only does this approach use the work shown in this chapter, but also the Maximum Diversity Problem (MDP) method of diverse sampling from Chapter 4. This approach has four major steps:

1. Curate sequences of an enzyme family of interest, then create a template set using the MDP approach.

2. Feed the latter template into a refined autoencoder and sample from the latent space X sequences.

3. Input X sequences into an improved filtering pipeline that selects Y potentially viable sequences, discarding the rest.

4. Create a panel set of diverse synthetic sequences using the MDP approach and characterise them in the laboratory.

This approach can also be iterative in nature, as any synthetic sequences proven to work can be fed backwards - to the training dataset of the network and the template sequences - resulting in further potential diversity at the next iteration. This approach can therefore, pending necessary refinement of the method, help increase the known diversity of enzyme families through artificial means.

5.4.3 Future work

An initial avenue of future work that could significantly improve the approach laid out in this chapter would be a more automated optimisation of the neural network training process. In particular, tuning parameters like the number of convolutional layers, trying out different optimisers and batch sizes, etc., could result in a better autoencoder for the purpose of generating novel enzyme sequences. As was pointed out in the previous section, a further reduction of the dimensions of the latent space would also be necessary to avoid the likely overfitting that occurred at training. Relatedly, provided higher levels of computational power for training like GPU nodes on higher performance computing servers become available, the depth of the autoencoder could be increased to the levels of deep learning, to allowing reduce the dimensions of the bottleneck layer, potentially improving the feature vector that is learned by the autoencoder.

Future work would not just concern the neural network structure and parameters, but also the data it is trained on. While retrieving all of the enzymes from Swiss-Prot created a dataset of over 200 thousand samples, which is substantial, there are ways of increasing this dataset size by multiple magnitudes. For example, including enzymatic sequences from TrEMBL, while a little risky in terms of quality of data, can be done in a safe way by adding high-similarity non-redundant subsets of it like UniRef90 [191]. Such a change could easily increase the training dataset size to factors of millions rather than hundred-thousands. Also, sequences below a certain size are likely to be fragments rather than full sequences, which could reduce the quality of the neural network. Filtering these out could therefore be another step forwards. In terms of sequence bias, one could also help reduce it in the training dataset using tools like cd-hit [192].

Finally, the last major avenue of future research identified is related to the generation of the novel sequences from the trained autoencoder. While in this work the latent space was sampled from using a gaussian distribution, other methods like training an entirely new neural network with the latent space of a template as input could yield better results. Also, it remains to be seen in more detail how much the size and diversity of a template set affects the diversity and quality of the generated sequences. While in this work the template set was split into two for phylogenetic reasons to provide guaranteed spread, a bigger more representative set for the sequences as a whole might have different qualities. Furthermore, with the revelation that the current autoencoder model overly produces perfect simple amino acid repeats in its sequences, an extra step to the filtering pipeline that scans sequences for such repeats and discards those above the expected amount could be significant.

5.4.4 Conclusions

In this study, it was shown that the generation of novel enzyme sequences using a discriminant autoencoder structure has strong potential as an approach for artificially increasing the diversity of enzyme families. They were shown *in silico* to be highly similar to native template sequences, and with further development can help increase the functional and physiological profiles of enzyme families in a more accessible way, due to the lack of a sequencing prerequisite for this approach.

6

Tools for an iterative laboratory-based approach to the exploration of enzyme families

Contents

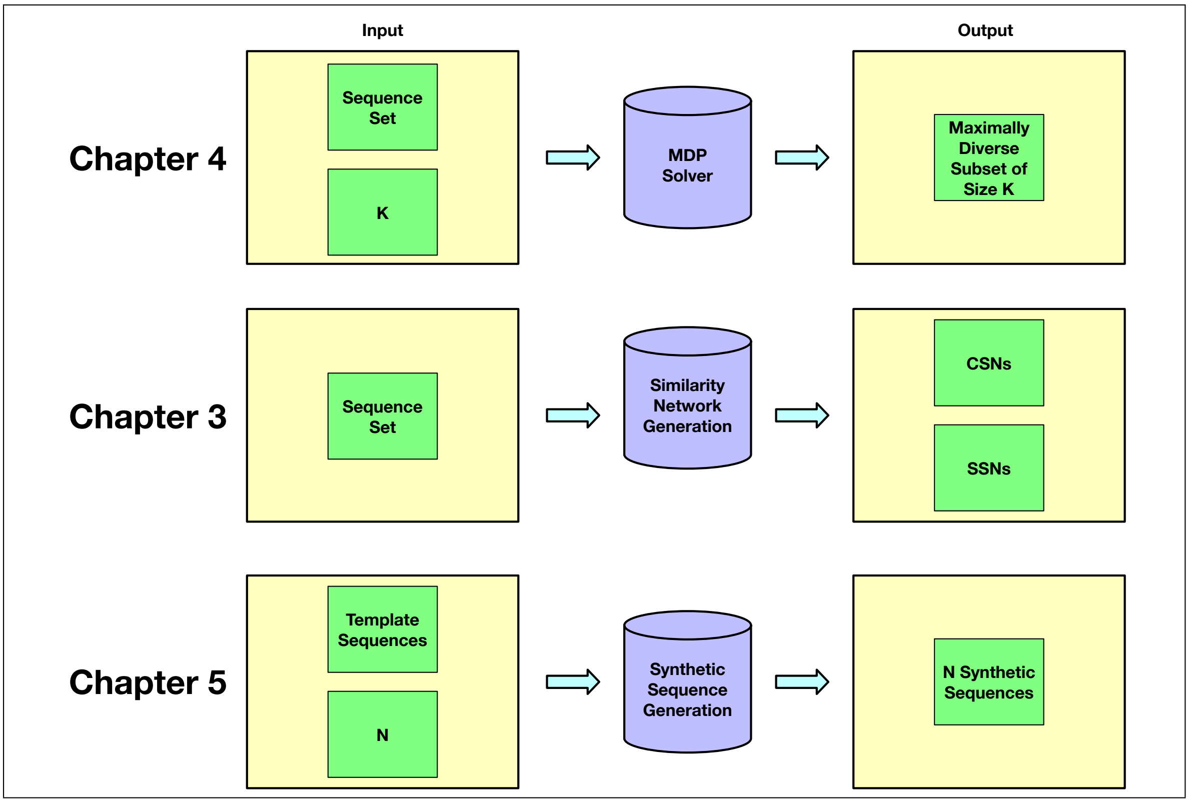


Figure 6.1: Figure showing the three main contributions of this thesis thus far.

6.1 Introduction

In the last three chapters, three methods were introduced for facilitating the generation of enzyme family panels (Figure 6.1):

1. Coevolution similarity networks (CSN), which are a novel type of network based on patterns of coevolving residues to complement sequence similarity based networks, in Chapter 3.
2. Sequence-based subset selection from diverse enzyme datasets based on solving the maximum diversity problem (MDP), in Chapter 4.
3. Generation of structurally and functionally viable synthetic sequences using neural networks, in Chapter 5.

The methods in Chapters 3 and 4 help create panels for enzyme families of interest, either by automatically selecting diverse subsets in the latter, or by revealing novel functional groupings in a family in the former. The machine learning-based method of

Chapter 5 attempts to add artificial diversity to enzyme family panels by generating synthetic enzymes based on a curated template set. Therefore, all approaches discussed so far exist to improve the panel curation process, resulting in sets of enzymes that are likely to display high amounts of catalytic diversity. Optimising this process was the principle aim of the research presented in this thesis, with the goal of positively impacting the diversity of panels of novel biocatalysts that are taken to be characterised in the laboratory.

For example, from a mined set of metagenomic enzymes of a family of interest, a subset of enzymes can be selected by solving the MDP (Chapter 4), followed by a polishing of the selection through an analysis that uses SSNs and CSNs (Chapter 3). This subset can then be complemented with a panel of synthetic enzymes that was generated using the machine learning-based approach introduced in Chapter 5. The endpoint of a panel curated in this manner is to be assayed experimentally, as the diversity of the panel can only be verified in the laboratory.

Characterisation assays performed on panels of enzymes help reveal knowledge about their potential functions, whether an assayed activity is positive or negative. Also, it is estimated that the number of discoverable and useful biocatalysts in metagenomic samples ranges from 1.4 to 19 enzymes per million base pairs [30]. The amount of diversity present in metagenome-scale data is therefore likely to be larger than the amount of diversity that can be represented in a single curated panel.

Furthermore, all methods discussed thus far can exploit novel knowledge produced by characterisation assays. For example, characterised enzymes can be annotated with their revealed functional profiles, which can help with the interpretion of similarity networks for future panels. Tested enzymes could also be filtered out for future rounds of MDP-based selection, increasing the chances that novel sequence space will be represented in the next panel. Finally, autoencoder-generated enzymes proven to be functional could either be used as an additional source of training data for the neural network, or to add further diversity to the template set. Therefore, the study of enzyme families through panel creation and characterisation approaches would benefit from being an iterative process in the long-term.

6.1.1 Tooling gaps in iterative enzyme family exploration

A framework for such an iterative approach to enzyme family exploration that includes experimental assays can be formulated, using metagenomic data as an example. This framework is divided into four main steps (Figure 6.2):

1. The first step is dataset building. which consists of compiling a set of sequences that are likely to be of some enzyme family. This step can be achieved using tools like *hmmsearch* [70] or BLAST [40] as discussed in section 2.2.2 of the Background.

2. The second step is panel preparation. A catalogue of sequences from the built dataset is then built by using the analysis tools described in previous chapters, such as MDP solvers (Chapter 4) and similarity networks (Chapter 3). Synthetic sequences generated by the autoencoder (Chapter 5) can also be added to the catalogue for extra artificial diversity.

3. The third step is laboratory characterisation. these sequences are characterised in the laboratory, passing through the synthetic biology workflow of design-built-test, producing metadata on the way.

4. The fourth and final step is learning. One can iterate on the next panel based on what was learned from the characterisation assays, which is the learn stage.

However, it is not currently accessible to use this framework due to gaps in tooling within and between individual steps (Figure 6.2). While every step can currently be performed individually, the framework is not optimised due to a lack of appropriate integrated tooling and storage platforms. For example, dataset building using multiple metagenomes requires the recording of provenance for individual enzyme sequences and metagenomes. Also, panel preparation methods are currently only accessible from the command-line, and are not integrated to a platform that links to the enzyme datasets built in step 1.

Furthermore, as discussed in section 2.5.3 of the Background, there is a lack in tooling for the build and test stages of the synthetic biology lifecycle. Therefore, as enzyme

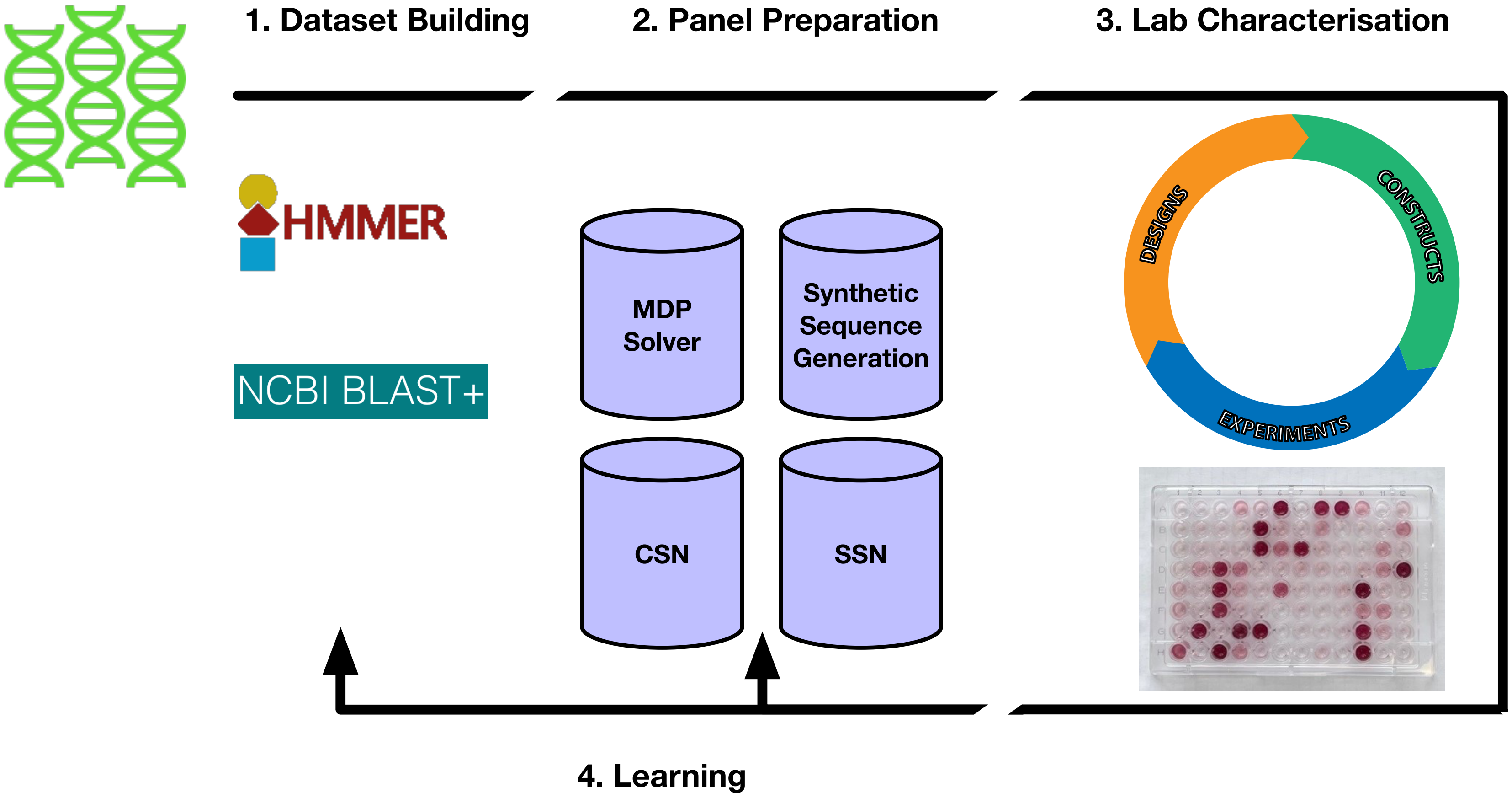


Figure 6.2: A workflow for an iterative approach to the characterisation of enzyme families, divided into four main steps. There are crucial gaps in between each steps, representing the lack of tooling needed to integrate the different processes of this workflow to make it accessible.

characterisation assays pass through these stages, the data and metadata produced does not have a bespoke storage platform to be stored on. Consequently, the datasets built in step 1 cannot easily be integrated with the newly discovered functional profiles of assayed enzymes. Finally, there is not currently a method of easily feeding back the results of these assays into a learning process to iterate further.

Crucially, every step produces substantial amounts of data and metadata, for both *in silico* and experimental steps, which need to be stored. For example, it is important to store the metagenomic sequences and the MDP-created subsets that are computed from them. Autoencoder-generated synthetic sequences need to be traced back to their original templates and the parameters used for their filtering. Furthermore, experimental protocols and the data produced by applying them, from gels to characterisation data of individually tested enzymes, also need to be stored. Finally, the synthetic biology lifecycle requires the storage of provenance links and other types of metadata to store, from construct design, to construct building, to raw experimental data.

The storage of these different data types and the relations between them is paramount to the success of the approaches discussed in this thesis over the long-term. The lack of integrated tooling that promotes an iterative approach to the characterisation of enzyme families complicates the framework further. There is therefore a need for tooling that fills these gaps, which was the second aim of this research.

6.1.2 Objectives

In this chapter, we describe two platforms to tackle the aforementioned gaps through these three objectives:

1. The integration of the *in silico* panel preparation pipeline and enzyme characterisation data.

2. The development of integrated platforms containing iterative functionality that guides future assays based on the results of previous runs.

3. The development of storage platforms for data and metadata produced by *in silico* and *in vitro* processes.

The first platform is Integrative Enzyme Lab (IntEnz-Lab), which is a bespoke web-interface that facilitates the laboratory-based exploration of enzyme sequences mined from metagenomes. It does so by integrating the creation of diverse enzyme panels and the assays that characterise them. This integration allows for an iterative approach towards exploring such datasets where the next round of assays is informed by previous results.

The second platform is SynBioHub-Lab, which is a repository for storing data about the different steps of the synthetic biology lifecycle described in section 2.5.1 of the Background - from design, to build, to test. Characterisation assays will pass through every step of this lifecycle, producing data that can be uploaded to SynBioHub-Lab, such as plasmid designs, built constructs, lab protocols, and experimental results.

Attribution: *SynBioHub-Lab was developed in collaboration with Dr. James McLaughlin. In particular, the initial code refactoring necessary to transition from using sboljs [124] to sbolgraph, and from JavaScript to TypeScript, are fully attributed to him.*

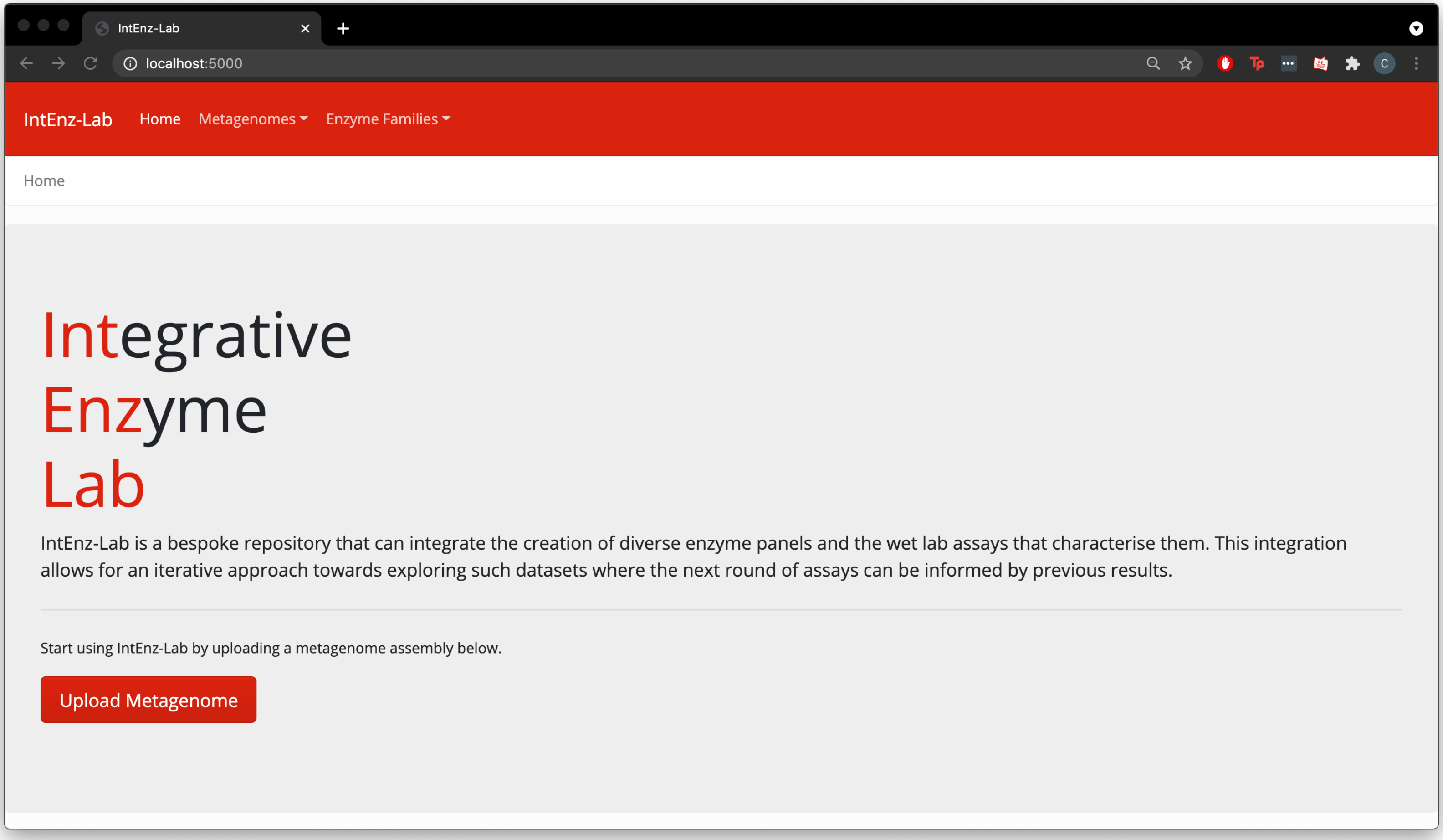
6.2 IntEnz-Lab

6.2.1 Background

IntEnz-Lab (Figure 6.3) is a bespoke repository that was developed to integrate enzyme family dataset building from metagenomic data, with the MDP-based creation of diverse panels from said datasets. IntEnz-Lab also integrates these steps with the characterisation data produced in the laboratory for said panels. A principle novel feature of IntEnz-Lab' is to use this integration to promote an iterative approach to exploring enzyme families by guiding future enzyme panels based on what has been characterised previously. The requirements of such a platform are fourfold:

1. A reliable database that connects enzyme family datasets to individual enzyme sequences and the metagenomes they originate from.

2. Producing subsets of user-defined sizes that are likely to contain high catalytic diversity using the MDP-based methods of Chapter 4, for the purpose of experimental characterisation.



Integrative
Enzyme
Lab

IntEnz-Lab is a bespoke repository that can integrate the creation of diverse enzyme panels and the wet lab assays that characterise them. This integration allows for an iterative approach towards exploring such datasets where the next round of assays can be informed by previous results.

Start using IntEnz-Lab by uploading a metagenome assembly below.

Figure 6.3: Screenshot of the front-page of IntEnz-Lab, which is a bespoke web-interface that facilitates laboratory-based exploration of enzyme sequences mined from metagenomes.

3. Integrating the aforementioned database with results of characterisation assays performed in the laboratory.

4. Assay-based sequence masking options that further narrow down the sequence space to be explored in future panels.

IntEnz-Lab was built with these requirements in mind as an approach to the laboratory-based iterative exploration of enzyme families and the diversity they contain.

6.2.2 Architecture of IntEnz-Lab

IntEnz-Lab is a platform consisting of two main services (Figure 6.4):

1. A web-interface that consists of both a frontend and a backend, with users interacting with the former. The backend populates the webpages, and validates and sends forms. Some forms lead to the running of workflows; from workflows that run *hmmsearch* [193] for dataset building, to the MDP-based panel selection.

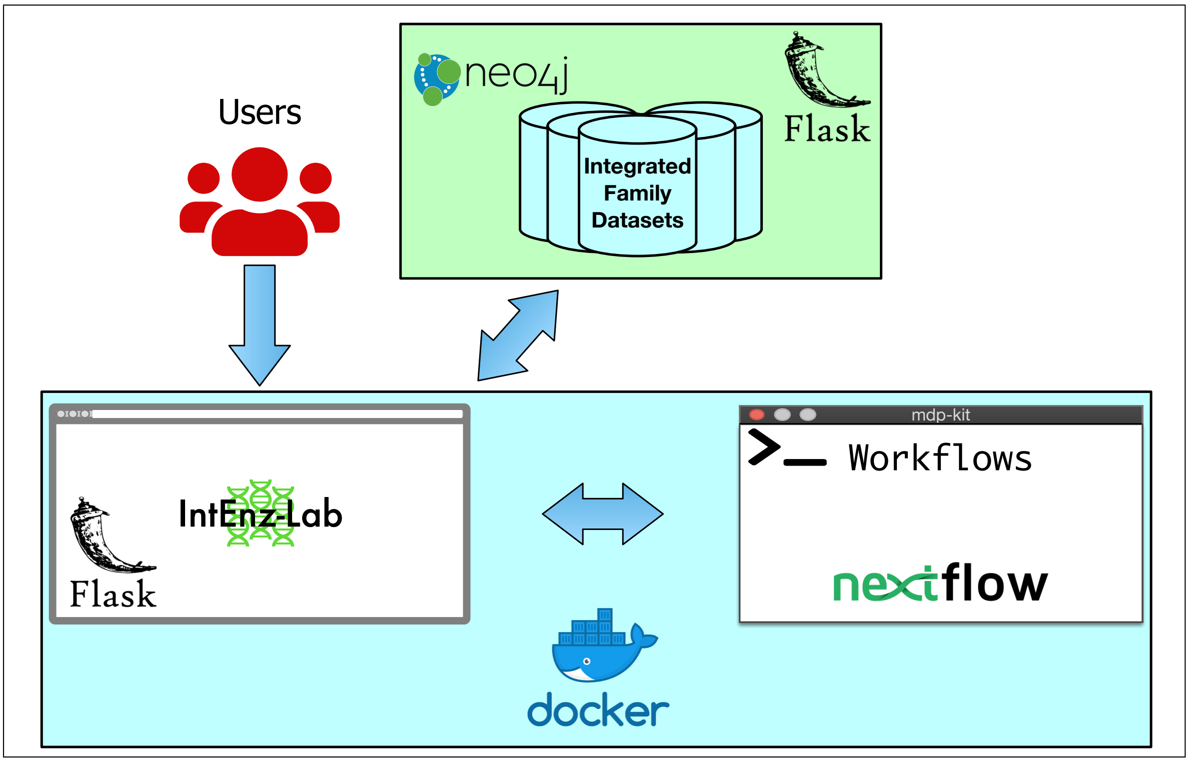


Figure 6.4: The architecture of IntEnz-Lab. IntEnz-Lab is made up of two main services: a web-interface that users interact with containing all of the necessary forms for performing its functionality, such as workflows, and graph database that uses Neo4J and that communicates with the back-end of the web-interface using a REST API.

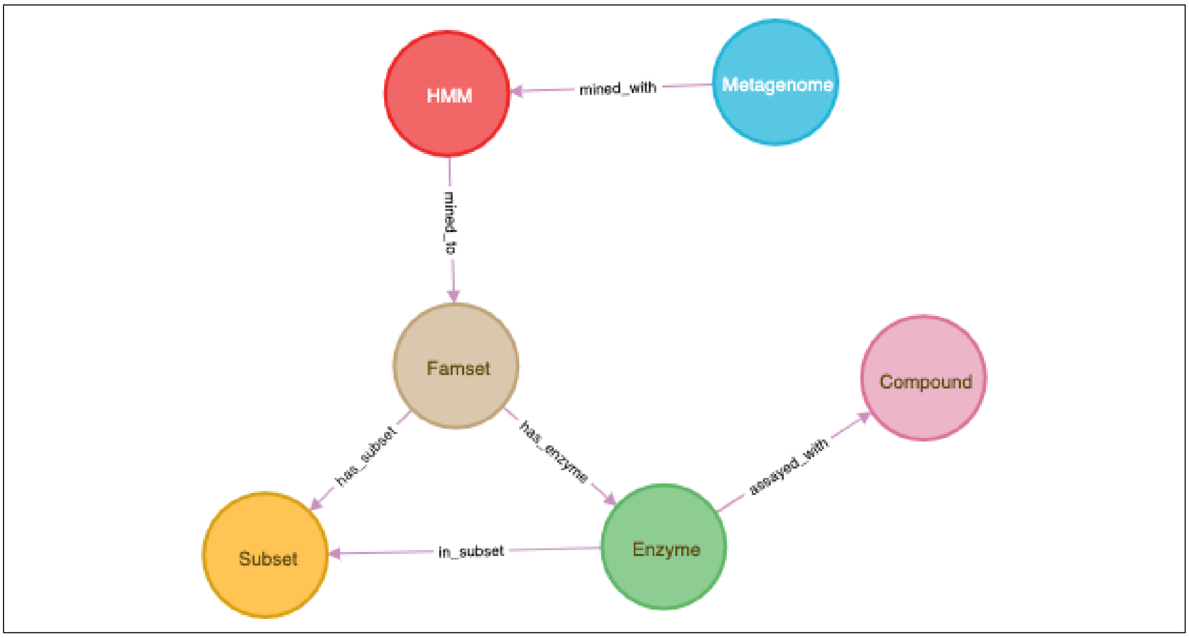


Figure 6.5: Database schema for the Neo4j graph database implemented for IntEnz-Lab. It is made up of six entities and six relationships, which can be seen in more detail in Figure 6.6.

2. A graph database that interfaces with the backend of the web-interface using a bespoke REST application programming interface (API).

The webpages and endpoints of both the frontend and backend of the interface are run entirely using Flask [194] and associated Python libraries. The workflows callable using IntEnz-Lab are either packaged up in the Nextflow scripting language, or are compiled binaries of software developed by others. Both the web-interface and the collection of workflows are containerised using Docker [195] for software isolation and ease of distribution.

6.2.3 Database Schema

IntEnz-Lab's graph database uses Neo4j [196], a NoSQL database management system (DBMS) based on graph storage, in contrast to traditional relational databases which are based on the storage of tables and their relations.

The schema uses six different entities and six different relationships. All six entities have both id and name properties, with the id functioning as a primary key. The schema can be seen in graph format in Figure 6.5. The detailed entity-relationship diagram of the schema in use by this database can be seen in Figure 6.6.

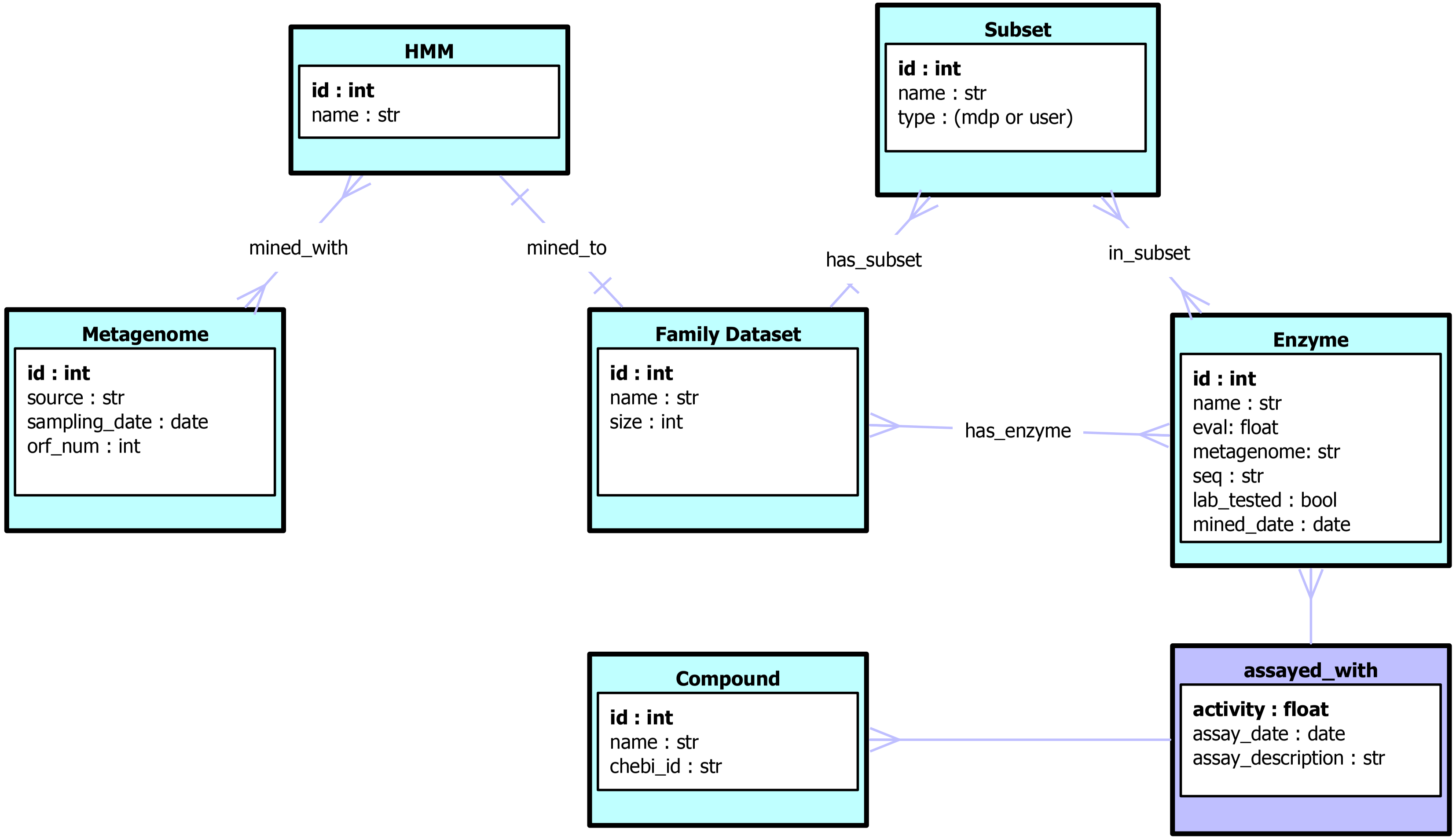


Figure 6.6: Entity-relationship diagram of the graph database in use by IntEnz-Lab. There are six main entities, all of which have id and name properties, with the former being the key property.

The first entity is *Metagenome*, which represents metadata about metagenomic assemblies uploaded to IntEnz-Lab, such as the physical source of the metagenome, its sampling date, and the amount of ORFs identified on it.

The next entity is *HMM*, which represents profile HMMs representing individual enzyme families of interest. This entity is needed as *hmmsearch* is used by IntEnz-Lab for dataset building.

Thirdly, the *Family Dataset* (or *Famset* for short) entity represents the individual datasets of enzyme families of interest. In terms of architecture, this entity plays the wrapper role of connecting three other entities in a way that makes logical queries easier. For example, each *Famset* has precisely one *HMM*, which can have *mined_with* relationships to many other metagenomes, allowing us to formulate queries that identify all of the metagenomes that were mined to create some family dataset.

The fourth entity is *Enzyme*, which represent individual mined enzyme sequences. Information about each *Enzyme* in the database includes peptide sequence, *hmmsearch* e-value, a reference to the originating *Metagenome*, and importantly a boolean property for whether the enzyme has been characterised yet in the laboratory. *Enzyme* entities have a relationship to *Famsets*, so that all of the hits of a certain family stored on the platform can easily be retrieved.

The fifth entity, *Subset*, is also connected to the *Enzyme* entity. It represents subsets or panels produced by runs of the MDP algorithm. The relationship between this entity and *Enzyme* makes it easy to find the members of any subset, and vice-versa whether an enzyme already belongs to any generated subsets. This entity is also connected to *Famset* to make it possible to find all of the subsets produced for a given family.

The final entity, *Compound*, represents chemical compounds that are assayed with *Enzyme* entities for activity, resulting in *assayed_with* relationships between them. *Compound* entities have properties that make it easy to identify the compound in question, such as the Chemical Entities of Biological Interest (ChEBI) ID and name. ChEBI is a standardised ontology for chemical entities [197]. The *assayed_with* relationship is the only one in this schema that has edge properties. These include assay activity, allowing IntEnz-Lab to quantitatively store and identify how reactive assayed enzymes were with connected chemical compounds.

6.2.4 Database Splitting

The database schema is such that there will be at least N *has_enzyme* edges, where N is the number of *Enzyme* entities in the database. If N were to get large enough - which is possible with high amounts of different families and metagenome mining - scaling could become an issue.

Therefore, to future-proof the database against this potential issue, the database is split into different graphs, one for each *Famset*. This change guarantees that only one enzyme family's worth of nodes ever has to be loaded into memory at one time, and the only other duplicate nodes possible are *Metagenome*, and *Compound* entities, which are relatively negligible in number. In queries, the argument used to refer to a specific *Famset*'s graph is *db_name*.

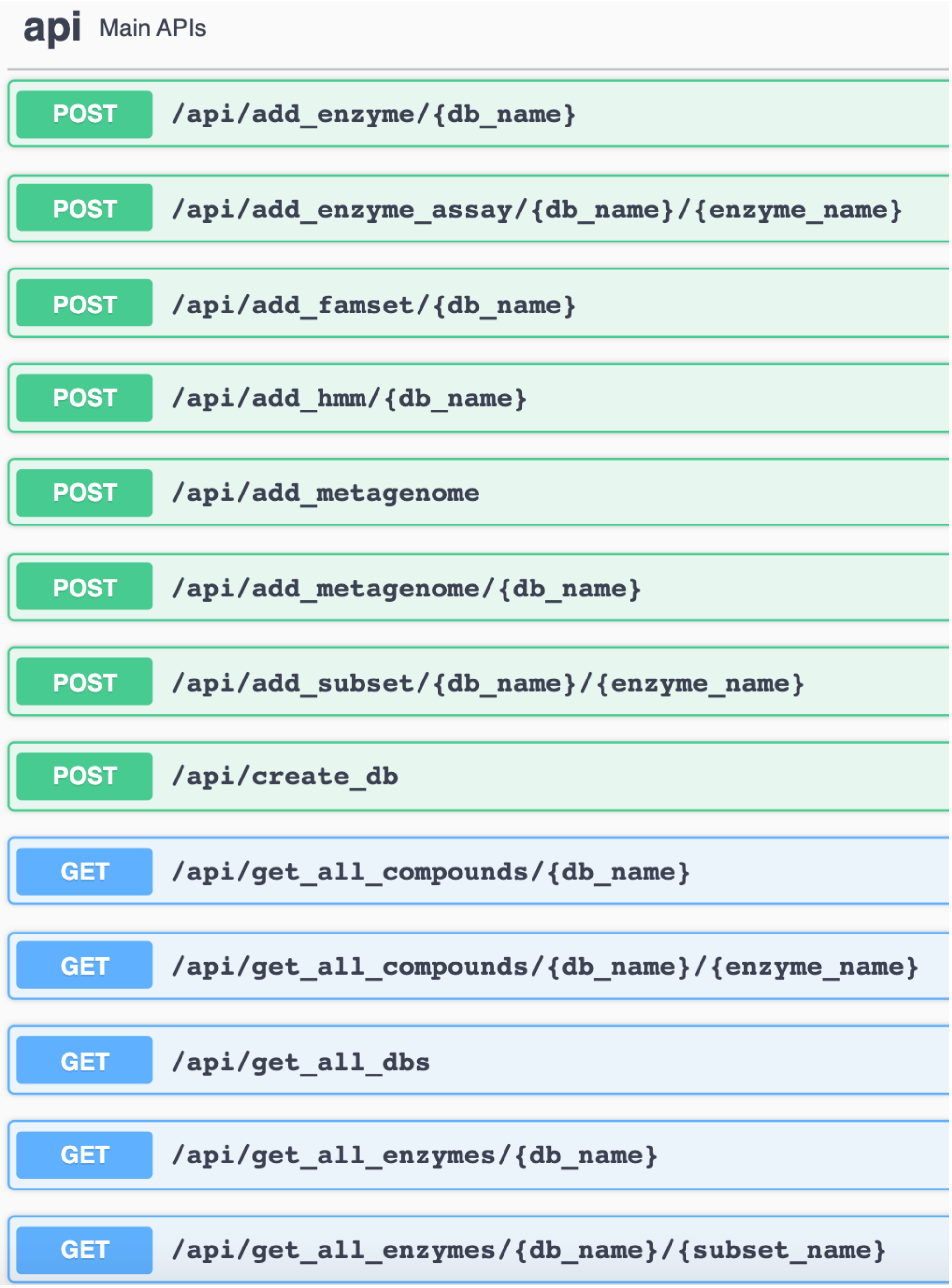
To allow for queries that retrieve all of the *Metagenome* and *Famset* entities in the database, along with the *mined_with* relationships that are present in the database, one further graph dataset called *metagraph* is created, where such entities and relationships are also stored.

6.2.5 Database API

The database API is made up of 20 endpoints, 12 of which are GET endpoints, with the remaining 8 being POST endpoints (Figure 6.7). In IntEnz-Lab, this API communicates between the Neo4j database and the backend of the interface. When any of the endpoints are called, the requests are first validated, then converted into a Cypher query to the Neo4j database.

For example, to retrieve all of the *Enzyme* nodes in the *aldo-keto-reductases Famset*, the */api/get_all_enzymes/*{*db_name*} endpoint is called with "aldo-keto-reductases" in place of *db_name*. This endpoint then performs the following Cypher query on the graph corresponding to the latter enzyme family:

```
MATCH (n:Enzyme)
RETURN n
```



api Main APIs

```
POST /api/add_enzyme/{db_name}
POST /api/add_enzyme_assay/{db_name}/{enzyme_name}
POST /api/add_famset/{db_name}
POST /api/add_hmm/{db_name}
POST /api/add_metagenome
POST /api/add_metagenome/{db_name}
POST /api/add_subset/{db_name}/{enzyme_name}
POST /api/create_db
GET  /api/get_all_compounds/{db_name}
GET  /api/get_all_compounds/{db_name}/{enzyme_name}
GET  /api/get_all_dbs
GET  /api/get_all_enzymes/{db_name}
GET  /api/get_all_enzymes/{db_name}/{subset_name}
```

Figure 6.7: Example GET and POST endpoints for the REST API developed for IntEnz-Lab.

This basic query returns all of the *Enzyme* nodes that belong to the *aldo-keto-reductases Famset*. The backend of the interface then processes the result of the query and presents it on the IntEnz-Lab interface.

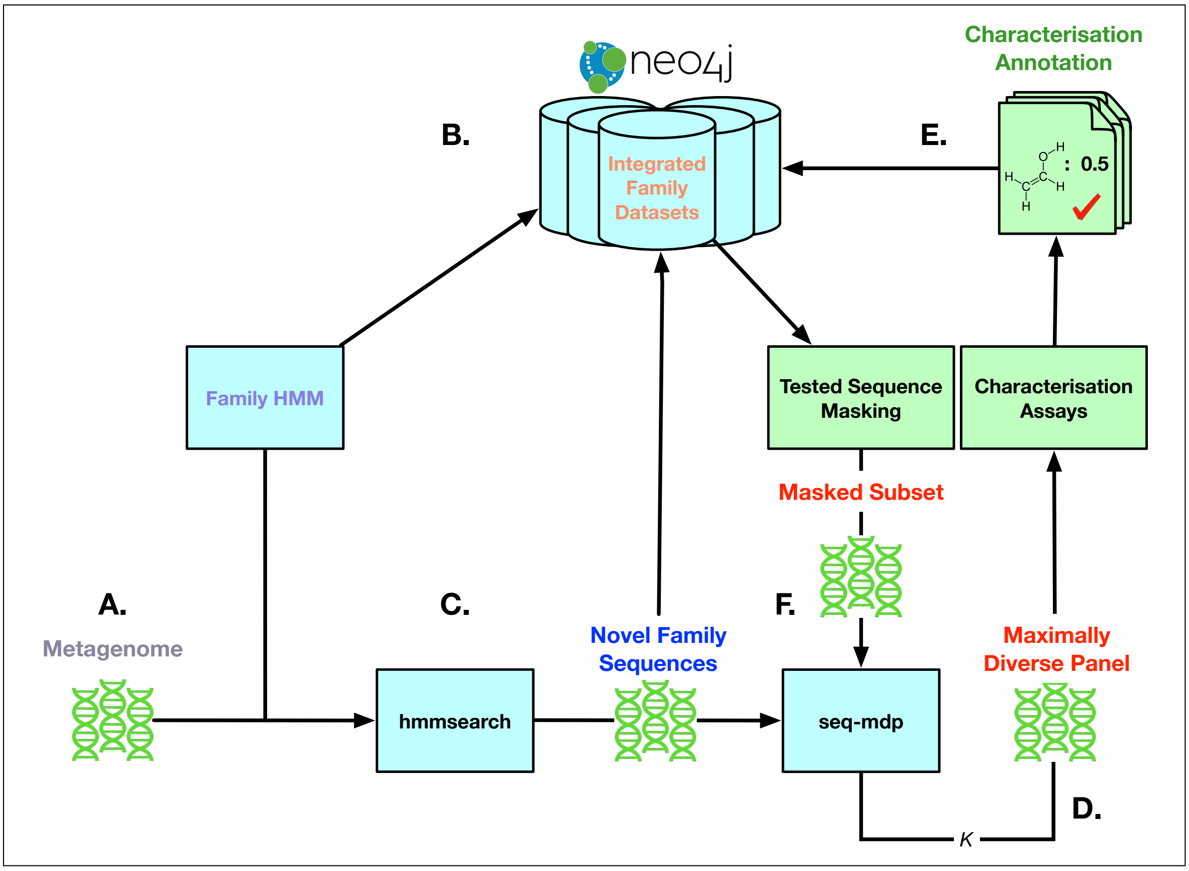


Figure 6.8: The main workflow and features available on IntEnz-Lab. There are six main features, which are A- uploading of metagenomic assemblies, B- instantiating a unified repository for metagenome-mined sequences of some enzyme family, C- the mining of a metagenome for some enzyme family, D- the sampling of diverse enzyme panels from hits using the MDP method (Chapter 4), E- the integration of characterisation data into the IntEnz-Lab database, and F- the masking of sequences from the database that were already characterised to guide future rounds of panel selection.

6.2.6 *Features of IntEnz-Lab*

IntEnz-Lab has six key features (Figure 6.8), which are the following:

1. Uploading metagenomic assemblies containing translated open reading frames (Figure 6.8-A).
2. Instantiating to the database a unified repository for novel enzymes of a particular enzyme family, using an HMM profile as a base (Figure 6.8-B).
3. Mining a metagenome for an enzyme family using its uploaded profile HMM (Figure 6.8-C). Mined sequences of this family are integrated with the unified

repository so that a large knowledge-base made up of the results of various mined metagenomes is built over time.

4. Sampling maximally diverse panels from metagenome-mined sequences by solving the MDP (Figure 6.8-D).

5. Uploading and integrating of enzyme characterisation data with the database (Figure 6.8-E).

6. Masking of already assayed enzymes that were integrated in the previous step for future rounds of MDP-based panel selection (Figure 6.8-F).

With these six functionalities, IntEnz-Lab is an accessible and centralised platform that integrates the data types and bioinformatics pipelines necessary for the iterative sampling and characterisation of putative enzymes mined from metagenomes.

6.2.7 An example walk-through of IntEnz-Lab

As IntEnz-Lab is driven by a web-interface, all of the features discussed in the precious section are easily accessible. This section goes through an example walk-through of IntEnz-Lab to show how such features are accessed.

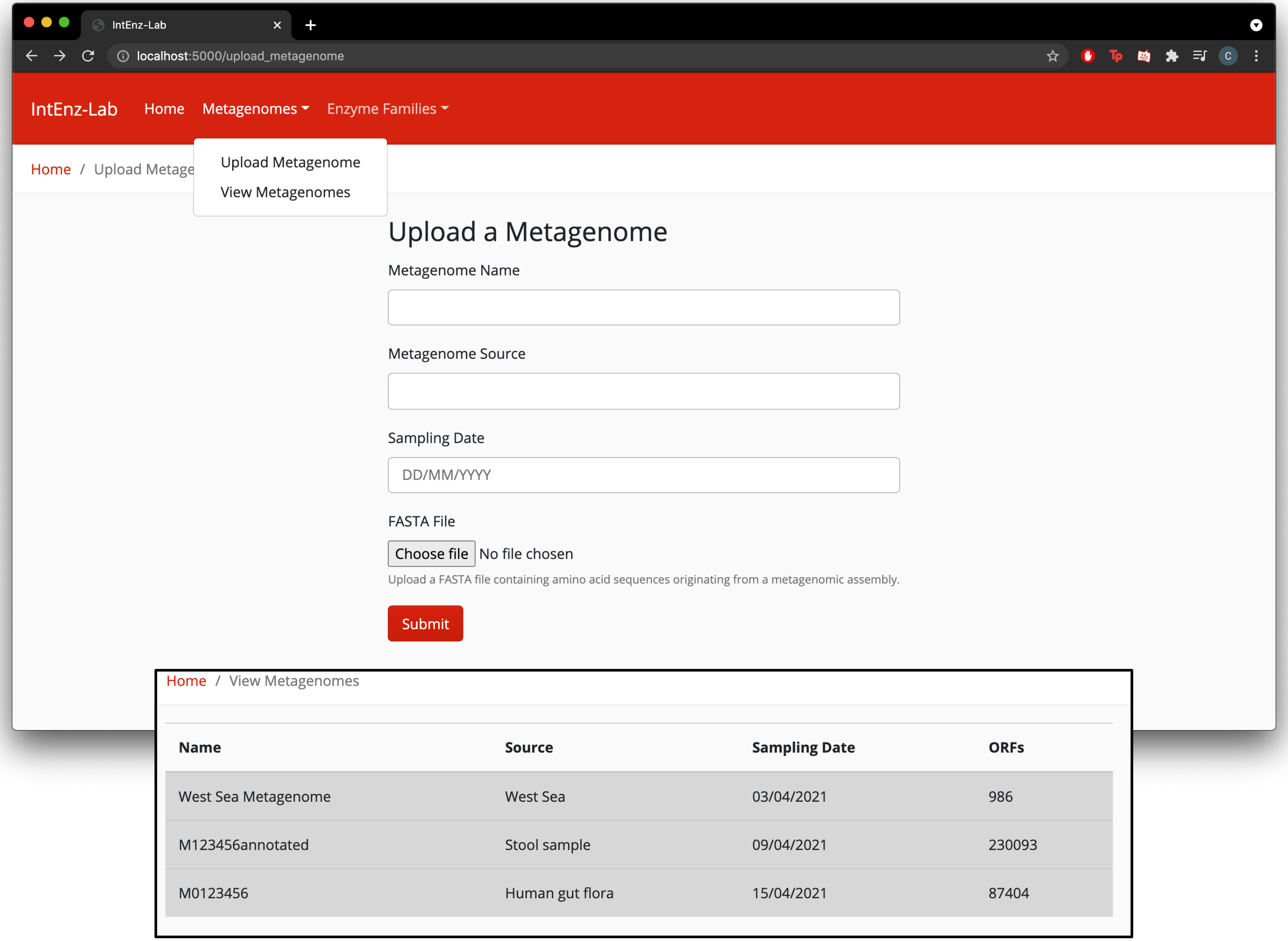
In Figure 6.3, the landing page of IntEnz-Lab can be seen. At the top, a navigation header can be seen, separating into two main tabs of pages - pages about metagenomes, and pages about enzyme families.

As was discussed in the preceding section, the first feature of this platform is the uploading of metagenomic assemblies. This feature is accessed in two ways: from the metagenome tab in the header, and a button at the bottom of the landing page, as it is chronologically going to be the first feature used. When either of these buttons are clicked, a form is presented to the user requiring four different input fields: metagenome name, metagenome source, sampling date, and an input FASTA file containing the sequences from the assembly. The first three of the latter fields make up three of the four properties of the *Metagenome* entity (Figure 6.6). Consequently, when the form is validated and submitted, a *Metagenome* entity containing is created with the inputted values in the *metagraph* dataset described in section 6.2.4. The next step of this walk-through is to the click on the header 'Metagenomes' tab, and then on 'View Metagenomes'.

The following page displays a table containing all of the *Metagenome* entities in the database along with relevant information about them (Figure 6.9). This information displayed consists of the fields uploaded by the user in the preceding form, except for one: ORFs. This number is the final property of the *Metagenome* entity, and is automatically generated from the uploaded assembly FASTA file by counting the number of sequences. This number is then be added to the newly created metagenome node in the database. These two pages conclude the first feature of IntEnz-Lab.

The next step in this walkthrough, and the second feature of the interface, is to instantiate a *Famset* for an enzyme family of interest. This feature is accessed by clicking on the 'Enzyme Families' tab and then on the 'Create Enzyme Family' button that presents itself. In a similar fashion to uploading a metagenome, a page containing a form for the user to fill is displayed (Figure 6.10). This form has three required fields: the name of an enzyme family, a description of it, and a file upload for an HMM file corresponding to the profile of the enzyme family in question. While uploading a metagenome only creates a node of entity type *Metagenome*, submitting this form creates multiple: a *Famset* node is added to the *metagraph* dataset, a new dataset named after the given enzyme family is created, and finally a *Famset* node and a *HMM* node are added to the latter dataset, with the latter two nodes connected by a *mined_to* edge. This step therefore instantiates a unified repository for mined enzymes of a certain family, which is quantitatively described using the uploaded HMM profile. The next step is to then click on the header 'Enzyme Families' tab, and then on 'View Enzyme Families'.

In a similar fashion to the table displaying existing metagenomes, the 'View Enzyme Families' page shows a table showing all of the created enzyme families thus far (Figure 6.11). Each entry has four columns: the name of the family, the given description, the number of hits, and the corresponding HMM. The first two and the final column all represent information input by the user for each family, while the number of hits is initialised at 0 as can be seen for the ADH family in Figure 6.11, which changes later



Upload a Metagenome

Metagenome Name

Metagenome Source

Sampling Date

DD/MM/YYYY

FASTA File

Choose file No file chosen

Upload a FASTA file containing amino acid sequences originating from a metagenomic assembly.

Submit

Home / View Metagenomes

Name	Source	Sampling Date	ORFs
West Sea Metagenome	West Sea	03/04/2021	986
M123456annotated	Stool sample	09/04/2021	230093
M0123456	Human gut flora	15/04/2021	87404

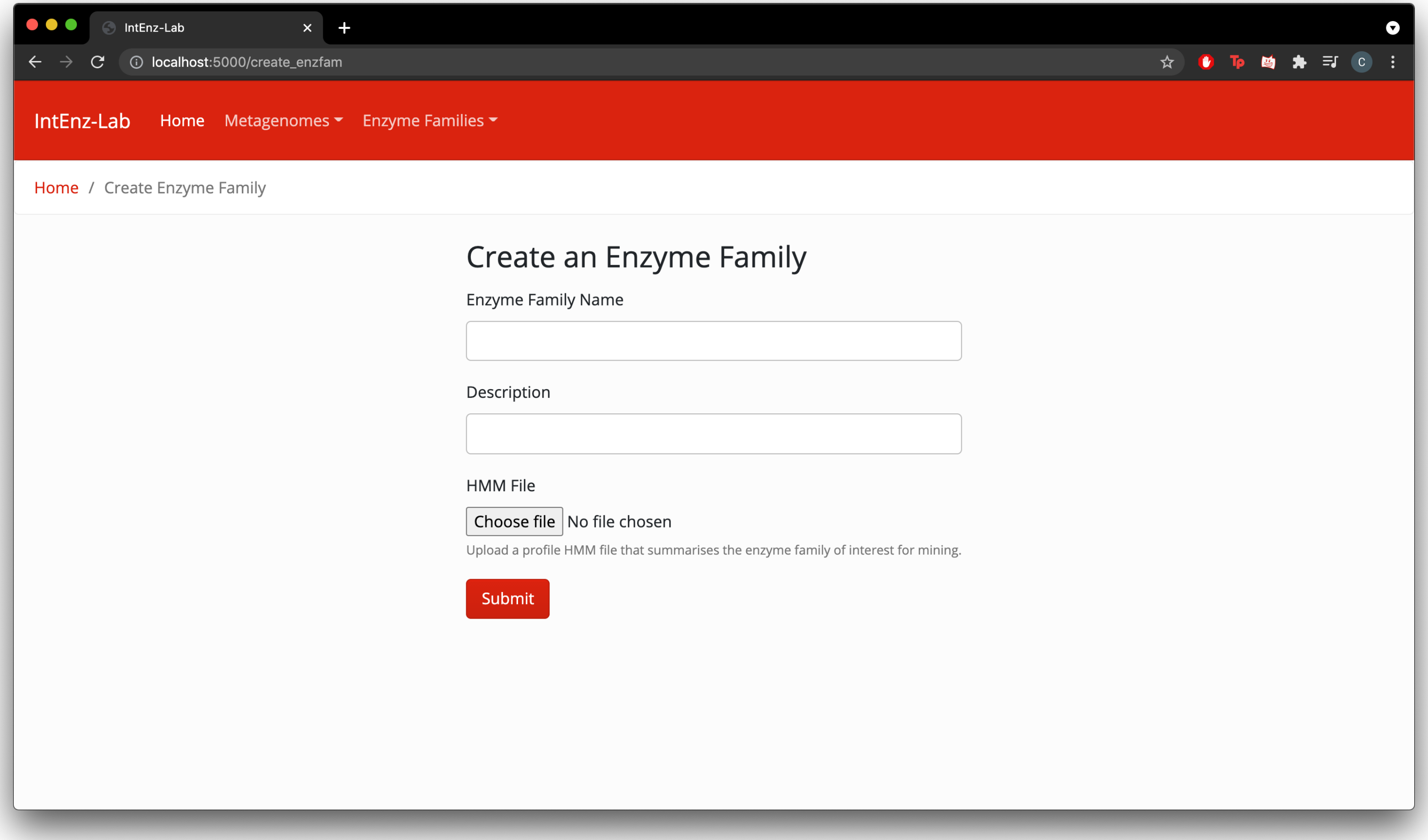
Figure 6.9: Screenshot of the IntEnz-Lab form that allows for the upload of metagenomic assemblies. As seen below, information about uploaded assemblies are displayed in a table in the 'View Metagenomes' page.

in the walk-through. The enzyme family names are clickable, leading to profile pages for each individual enzyme family (Figure 6.11). As the ADH family is initially empty, there is little to display on its page. However, each family page is presented with multiple clickable tabs, including the 'Actions' tab. This tab contains three different functions of IntEnz-Lab: mining a metagenome, creating a diverse panel, and adding an enzyme assay. These individual enzyme family pages are therefore the hubs of the platform, where pipelines can be run for enzyme families of interest.

The first action that can be performed for an enzyme family is the mining of a metagenome. Selecting a specific enzyme family page and then clicking on the 'Mine a Metagenome' button leads to another form page (Figure 6.12). This form is made up of a single field: a drop-down menu that is populated by all of the *Metagenome* entities created thus far. These action pages programmatically follow from individual enzyme family profiles, meaning that the context for which family a metagenome should be mined for is inherently known. Therefore, when this form is submitted, the HMM uploaded for a given family is located and used as input along with the selected metagenome to *hmmsearch* with a default e-value threshold of $1e^{-5}$.

In the backend, a node of type *Metagenome* is then created in the relevant family dataset, and connected to its *HMM* node by a *mined_with* edge. Following this, for every sequence hit resulting from the *hmmsearch* run, a *Enzyme* node is created with the appropriate values for its properties (Figure 6.6): the name and sequence is retrieved from the FASTA header in the metagenome assembly file, and the e-value is retrieved from the *hmmsearch* hit. The provenance metagenome property is also stored, and the current date is assigned as the *mined_date*. Also, the *lab_tested* property is initialised as *False*. Finally, for each of these new *Enzyme* nodes, a *has_enzyme* edge is created to the *Famset* node of the dataset.

With the database now populated with *Enzyme* nodes, the profile page for a mined enzyme family changes significantly (Figure 6.12). First, a list of metagenomes that were mined for this dataset are displayed. Also, a table is now populated by all of the mined enzymes, and information relevant to them is displayed such as the originating metagenome, the e-value for the hit, and whether they've been tested in the lab yet. This table is dynamic and can be sorted differently, depending on the column that is



Create an Enzyme Family

Enzyme Family Name

Description

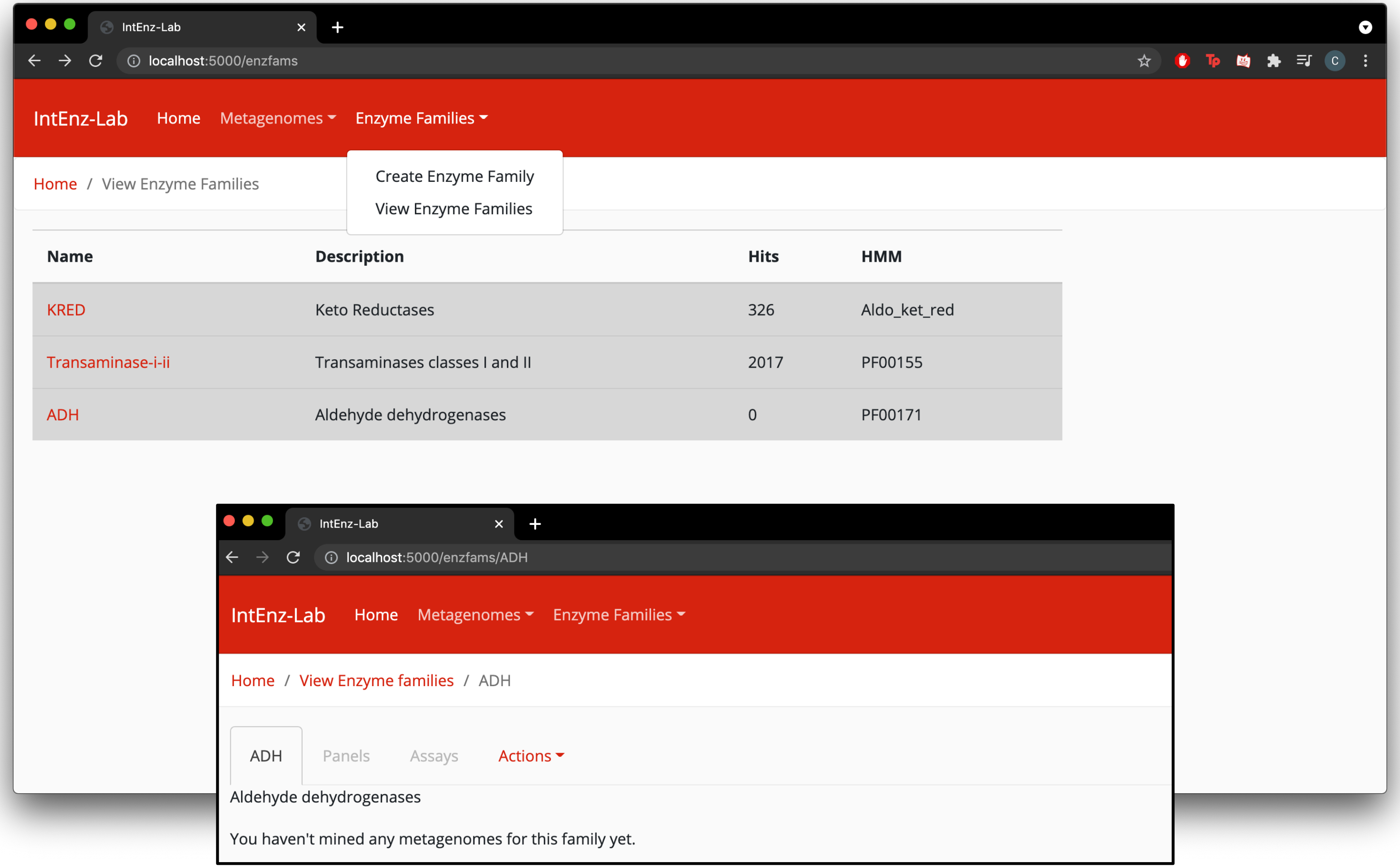
HMM File

Choose file No file chosen

Upload a profile HMM file that summarises the enzyme family of interest for mining.

Submit

Figure 6.10: Screenshot of the IntEnz-Lab form for creating an enzyme family entry. Such entries consist of unified repositories for any sequences mined from metagenomes using IntEnz-Lab, thanks to the required profile-HMM field in the form.



Name	Description	Hits	HMM
KRED	Keto Reductases	326	Aldo_ket_red
Transaminase-i-ii	Transaminases classes I and II	2017	PF00155
ADH	Aldehyde dehydrogenases	0	PF00171

Aldehyde dehydrogenases

You haven't mined any metagenomes for this family yet.

Figure 6.11: Screenshot of IntEnz-Lab showing the 'View Enzyme Families' page, which displays a table containing information about the currently existing enzyme families, including the number of hits and the HMM in use. Their names are clickable, leading to a profile page for each enzyme family. When first instantiated, the profile page is mostly empty as it has not been mined for sequences yet, with most functionality disabled until then.

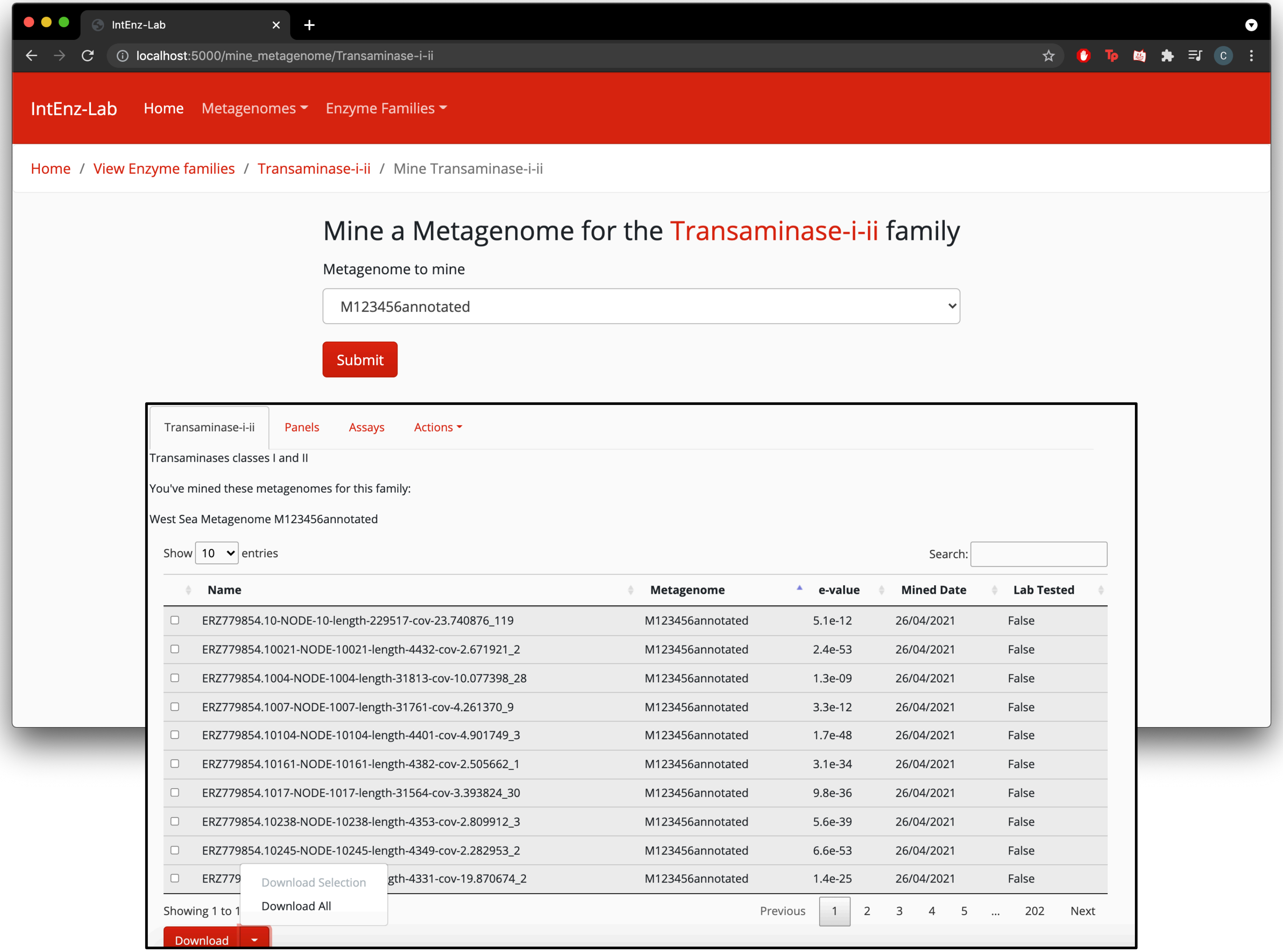
clicked, and is searchable using the search bar. Using the latter two functionalities, it is possible to download either all of the hits in the table, or a selection, using a checkbox associated with each row of the table. This table makes it possible to gather hits mined from *hmmsearch* and perform further bioinformatics analysis outside the platform, like alignments and trees, or to share the hits with others.

The next action and feature of IntEnz-Lab is to sample MDP-based subsets from a set of mined hits. Once again, starting at the hub that is an enzyme family page, clicking on the 'Actions' tab followed by the 'Create Diverse Panel' button leads to a new form page. This page only has two fields: the panel size, which is required, and a masking filter, which is optional. This form already contains the enzyme family context and it is therefore not needed as input.

Once a panel size is chosen and the form is submitted, the tabu search algorithm for solving the MDP discussed in Chapter 4 is run with all of the enzyme hits previously mined for this enzyme family, with the subset size K equal to the panel size chosen. Once the workflow is finished, a node of type *Subset* is created, and *has_enzyme* edges are made between it and all of the *Enzyme* objects sampled into the subset by the algorithm. A single *has_subset* edge is also created between the *Subset* node and the *Famset* node of the dataset. Once at least one such panel has been created, the 'Panels' tab in the enzyme family profile page is usable, and displays another table containing information about individual panels, such as size and the average e-value of the hits (Figure 6.13). Clicking on the automatically generated name of a panel downloads a FASTA file containing the sequences of the panel.

The optional masking filter is another novel feature of IntEnz-Lab. When used, it will discard enzymes previously marked as having been tested in the lab for the next run of the MDP workflow. This feature allows for the iterative sampling of diverse sequences for assaying to guarantee previously characterised sequences are not sampled again.

Finally, there is one more novel action that can be performed on enzyme families on IntEnz-Lab - the integration of characterisation data with the rest of the database. The final button in the 'Actions' tab of an enzyme family page is the 'Add Enzyme Assay' button, and when clicked another form is presented with just one field (Figure 6.14). The form requires a comma-separated values (CSV) file of a specific format, an example


IntEnz-Lab
localhost:5000/mine_metagenome/Transaminase-i-ii
IntEnz-Lab
Home
Metagenomes
Enzyme Families
Home / View Enzyme families / Transaminase-i-ii / Mine Transaminase-i-ii

Mine a Metagenome for the Transaminase-i-ii family

Metagenome to mine: M123456annotated

Submit
Transaminase-i-ii
Panels
Assays
Actions

Transaminases classes I and II

You've mined these metagenomes for this family:

West Sea Metagenome M123456annotated

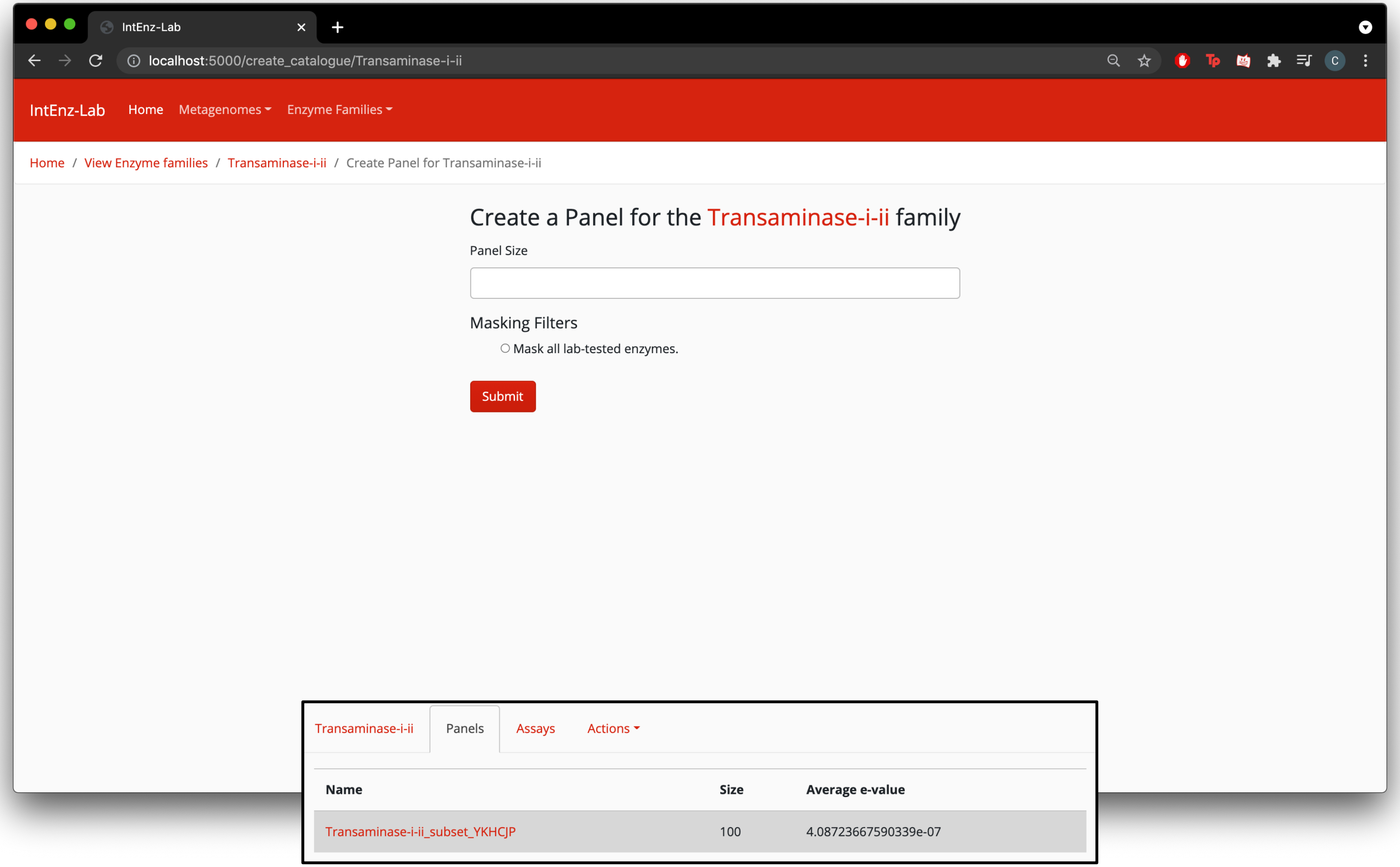
Show 10 entries

Search:

	Name	Metagenome	e-value	Mined Date	Lab Tested
□	ERZ779854.10-NODE-10-length-229517-cov-23.740876_119	M123456annotated	5.1e-12	26/04/2021	False
□	ERZ779854.10021-NODE-10021-length-4432-cov-2.671921_2	M123456annotated	2.4e-53	26/04/2021	False
□	ERZ779854.1004-NODE-1004-length-31813-cov-10.077398_28	M123456annotated	1.3e-09	26/04/2021	False
□	ERZ779854.1007-NODE-1007-length-31761-cov-4.261370_9	M123456annotated	3.3e-12	26/04/2021	False
□	ERZ779854.10104-NODE-10104-length-4401-cov-4.901749_3	M123456annotated	1.7e-48	26/04/2021	False
□	ERZ779854.10161-NODE-10161-length-4382-cov-2.505662_1	M123456annotated	3.1e-34	26/04/2021	False
□	ERZ779854.1017-NODE-1017-length-31564-cov-3.393824_30	M123456annotated	9.8e-36	26/04/2021	False
□	ERZ779854.10238-NODE-10238-length-4353-cov-2.809912_3	M123456annotated	5.6e-39	26/04/2021	False
□	ERZ779854.10245-NODE-10245-length-4349-cov-2.282953_2	M123456annotated	6.6e-53	26/04/2021	False
□	ERZ779 gth-4331-cov-19.870674_2	M123456annotated	1.4e-25	26/04/2021	False

Showing 1 to 1

Figure 6.12: Screenshot of IntEnz-Lab's ”Mine Metagenome” page. This page is accessible from one of the Action tabs shown in Figure 6.11, giving context to the form about which enzyme family is being mined, and therefore which HMM to use. *hmmsearch* is run using the respective profile-HMM on the metagenome that a user chose to mine. When this is done, the profile page of the mined family will now contain a table populated by information of all the resulting hits. New pages will now also be available in the Actions tab.


IntEnz-Lab
localhost:5000/create_catalogue/Transaminase-i-ii
IntEnz-Lab
Home
Metagenomes
Enzyme Families
Home / View Enzyme families / Transaminase-i-ii / Create Panel for Transaminase-i-ii

Create a Panel for the Transaminase-i-ii family

Panel Size

Masking Filters

○ Mask all lab-tested enzymes.

Submit
Transaminase-i-ii
Panels
Assays
Actions

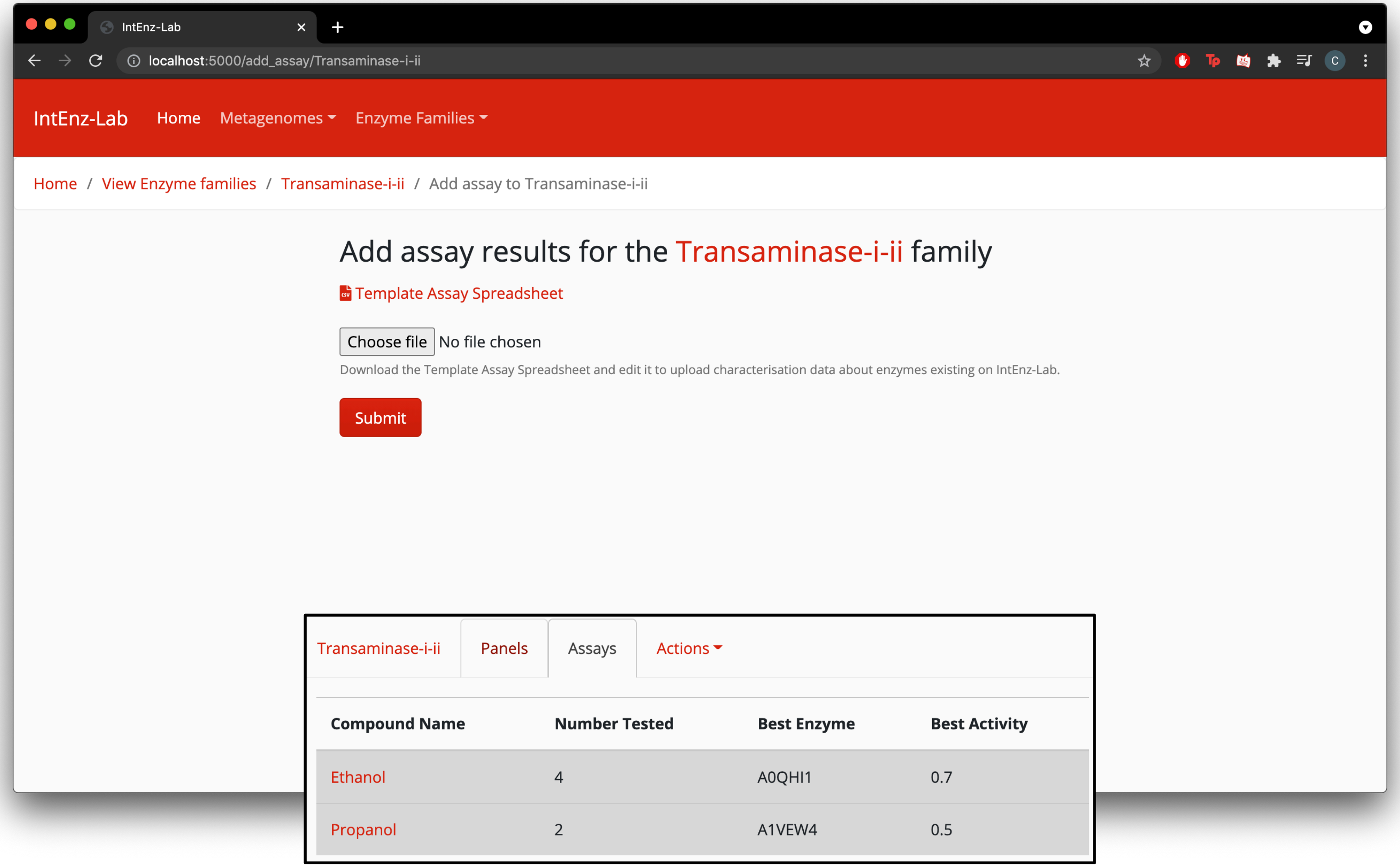
Name	Size	Average e-value
Transaminase-i-ii_subset_YKHCJP	100	4.08723667590339e-07

Figure 6.13: Screenshot of IntEnz-Lab of the 'Create Panel' form. Another page accessible from the Actions tab, a panel of a user-given size is created using the tabu search MDP method introduced in Chapter 4. This enables the Panels tab in an enzyme family's profile page, which displays information about any created panels.

of which can be seen in Table 6.1. This file will contain the raw characterisation data for enzymes tested in the laboratory, along with the necessary mapping column to integrate this data to the knowledgebase on IntEnz-Lab.

The CSV file should contain five columns, and at least two rows including the header, containing these column names: Enzyme Name, Compound Name, ChEBI ID, Activity, and Date, in that order. Each row corresponds to an individual assay, with the Enzyme Name mapping to its respective entry in IntEnz-Lab. Then, the Compound Name and ChEBI ID represent the chemical substrate that was tested for some enzyme, with the ChEBI ID linking to a standardised ontology for chemical entities [197]. Then, the quantitative activity level goes into the next column, followed by the date the assay was performed in the final one. As can be seen in Table 6.1 for A0PXP5, it is possible to integrate multiple chemical assays for the same enzyme in the same spreadsheet. To make filling this spreadsheet easier, IntEnz-Lab provides a template file with the examples shown in Table 6.1.

When such a form is validated and submitted, the backend of IntEnz-Lab will first create *Compound* nodes for every substrate in the spreadsheet that does not yet exist in the database. The ChEBI ID and name are taken from the spreadsheet and added to each node. Then, *assayed_with* edges are created between every tested *Enzyme* node and the compounds. This edge type is the only one in the IntEnz-Lab schema that has properties, including the substrate activity and the date the assay was performed. Once an assay has been added for the first time, the enzyme family profile page changes again - the 'Assays' tab becomes available, and a table displaying the different compounds tested is shown (Figure 6.14). This table displays the number of enzymes tested for each compound to provide users with information about which compounds have already been heavily tested. The atable also identifies the enzyme with the highest activity for some compound. This feature of integrating assay data with the rest of the database is crucial to making IntEnz-Lab a platform that facilitates the iterative exploration of enzyme families through wet lab characterisation experiments.



Home / View Enzyme families / Transaminase-i-ii / Add assay to Transaminase-i-ii

Add assay results for the Transaminase-i-ii family

Template Assay Spreadsheet

Choose file No file chosen

Download the Template Assay Spreadsheet and edit it to upload characterisation data about enzymes existing on IntEnz-Lab.

Submit

Compound Name	Number Tested	Best Enzyme	Best Activity
Ethanol	4	A0QHI1	0.7
Propanol	2	A1VEW4	0.5

Figure 6.14: Screenshot of IntEnz-Lab showing the 'Add assay' form. A template spreadsheet (Table 6.1) can be downloaded by the user to be filled and re-uploaded with characterisation data from enzyme assays. This data is integrated with the rest of the database, and the Assays panel becomes populated with a table showing information about the different compounds that have been assayed.

Table 6.1: Table showing the template spreadsheet for adding characterisation data to IntEnz-Lab.

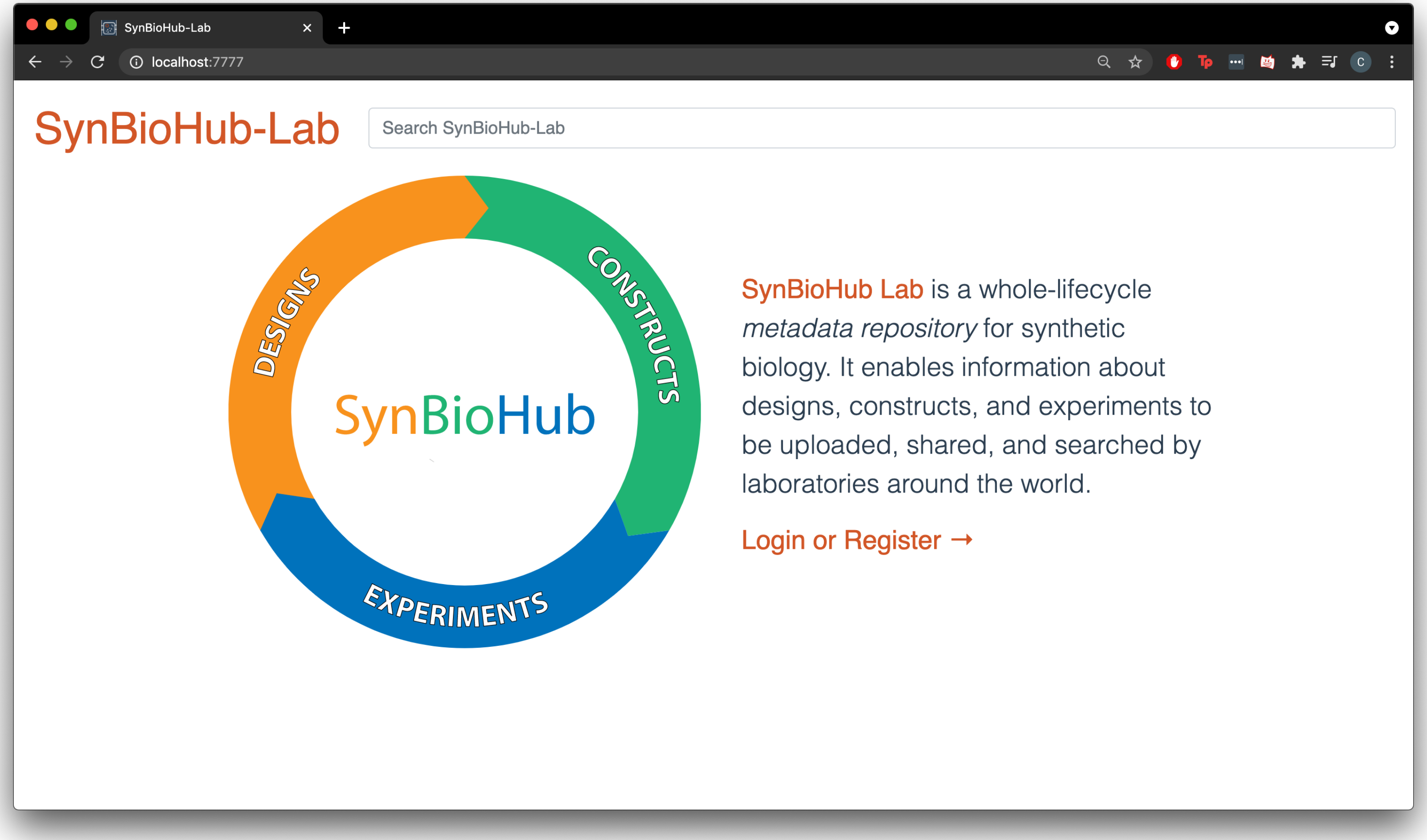
Enzyme Name	Compound Name	ChEBI ID	Activity	Date
A0PXP5	Ethanol	CHEBI:16236	0.5	2021-01-12
A0Q9F3	Ethanol	CHEBI:16236	0.3	2021-01-12
A0QHI1	Ethanol	CHEBI:16236	0.7	2021-01-12
A0PXP5	Propanol	CHEBI:28831	0.01	2021-01-12

6.3 SynBioHub-Lab

6.3.1 Background

SynBioHub-Lab is a partial rewrite of an existing platform called SynBioHub [109]. SynBioHub is an open-source repository for storing and sharing biological designs, and is meant to be a unified hub for synthetic biologists to upload their designs onto. As discussed in section 2.5.2 of the Background, the Synthetic Biology Open Language (SBOL) is the primary data model to record information on the four different steps of the synthetic biology lifecycle - design, build, test, learn - and is the format in use by SynBioHub's backend for manipulating and displaying uploaded designs. The laboratory-based work required for the exploration of enzyme families passes through the whole lifecycle, producing multiple different data on the way that need to be stored. The SBOL data model has already reached multiple important milestones in its attempts at modelling the synthetic biology lifecycle. Crucially for this work, the data model for the links between designs, constructs, and the respective experiments performed on them, was formulated and approved in 2019 [116].

However, while SynBioHub is undoubtedly a key resource for modern synthetic biology, and is well-maintained and updated by a team of developers from the SBOL community, using it to exploit the SBOL model's latter milestones is not yet accessible. In particular, while SynBioHub can easily display sequence designs, the nomenclature in use by the platform necessitates a significant level of knowledge about the SBOL data model to optimally use it. Similarly, the overall flow of control a user would undertake is one that follows overall SBOL logic. For example, everything on SynBioHub belongs to a *Collection*, which is the class defined by SBOL as a set of other SBOL items. Whether you want to search for entries, or upload new ones, a collection



SynBioHub Lab is a whole-lifecycle *metadata repository* for synthetic biology. It enables information about designs, constructs, and experiments to be uploaded, shared, and searched by laboratories around the world.

Login or Register →

Figure 6.15: Screenshot of SynBioHub-Lab's front-page, which is a metadata repository for the synthetic biology lifecycle geared for experimentalists.

must first be created. This problem becomes particularly amplified for SynBioHub's implementation of the next two steps of the lifecycle - build and test - as these are performed in the laboratory, usually by researchers whose interests do not align with the need for working knowledge of the complex SBOL model.

While recent changes to the data model have simplified it significantly [115], it is still relatively unfriendly to understand without specialist domain knowledge of SBOL, as proven by the drive to create layers of abstraction that help more users interact with SBOL, such as ShortBOL [123]. Given that the nomenclature and interface design in use by SynBioHub is SBOL-heavy, it is therefore difficult to market towards to experimentalists. Consequently, synthetic biology experiments that are past the design stage often do not have a structured repository to store resulting data and metadata that corresponds to their needs. Indeed, a review of the current state-of-the-art platforms for managing synthetic biology data ruled that SynBioHub, while useful, should be more ”biologist-friendly and hide the underlying resource description framework (RDF) predicates” [198].

For the purposes of this work, while storing a plasmid design containing enzyme sequence to be characterised can be done on SynBioHub, it is harder to keep track of information relevant to the built construct e.g. host context, lab protocol used, metadata about who built it and when. A similar problem arises for the test stage; when an enzyme is characterised, similar protocols and metadata need to be stored, on top of needing to record information like the construct(s) used, location of raw experimental data. The knowledge-floor required for handling this data should of course be ”biologist-friendly” from start to finish, unlike on SynBioHub [198].

Therefore, there is strong potential in a new platform that allows for SBOL-based storage of these different data in a way that is designed for use by experimentalists, not just in the nomenclature and levels abstraction over the SBOL data model, but also in the way the site is designed. The result of this endeavour is the focus of this section, and is a partial rewrite of SynBioHub called SynBioHub-Lab; a metadata repository for the different steps of the cycle (Figure 6.15). SynBioHub-Lab tries to tackle the aforementioned challenges through the following two major changes:

1. A set of nomenclature changes, swapping the SBOL-heavy lexicon in-use on SynBioHub with terms more widely known to experimentalists.
2. A redesign of the front-end, both in visuals and in the flow of control undertaken by users.

The next three sections dive into these changes. Importantly, SynBioHub-Lab was developed when the latest release of SBOL was 2.3, and therefore does not include additions and features that were added in 3.0 onwards [115].

6.3.2 Nomenclature changes

Due to the relative complexity of the nomenclature in use by SynBioHub, all references to SBOL-specific terms were swapped for terms more geared towards experimentalists in SynBioHub-Lab. More specifically, a dictionary was compiled from discussions with a multitude of laboratory professionals for a series of important synthetic biology terms. This dictionary was then applied to the nomenclature on SynBioHub-Lab.

The meaning and purpose of the terms replaced are described in-depth in section 2.5.2 of the Background, and are the following:

- The term *Collection*, which represents lists of other SBOL items, was replaced with *Project*.
- The term *ComponentDefinition*, a subclass of the *TopLevel* class that represents sequence designs in SBOL, is replaced with the term *Design*.
- The term *Implementation*, a subclass of the *TopLevel* class that represents built constructs along with their provenance links to the design stage, is replaced with the term *Construct*.
- Design-stage components can be of multiple different types, like promoters, proteins, ribosome binding sites, etc. In SynBioHub-Lab, these components are explicitly referred to as their respective component type.

- SBOL uses the ontology Prov-o [119] to record provenance. The Prov-o term *Agent*, which represents the person(s) and/or tool(s) that generate some *TopLevel* object, is replaced by *Researcher*. While researchers are only a subset of what an *Agent* can be according to SBOL, it is a conscious decision to narrow it down to a definition more helpful to SynBioHub-Lab.

- SynBioHub uses the *TopLevel* Prov-o class *Plan* to refer to the steps undertaken by some lab activity. In SynBioHub-Lab, these are instead directly referred to as lab protocols.

- SynBioHub uses the predicate *wasGeneratedBy* and *wasDerivedFrom* to connect relevant *TopLevel* objects to *Agents* and other *TopLevel* objects, respectively. Predicates like these are never displayed in SynBioHub-Lab, instead directly referring to the object the predicated point to, like *Researchers* for *Agents*, but also *Original Design* instead of a *wasDerivedFrom* predicate linking to a *Construct*'s respective design.

The terms used in SynBioHub-Lab were chosen based on discussions with specialists, but the difference in reach for the new lexicon can also be shown quantitatively for some of the replacements. On Google scholar [199], searching for 'ComponentDefinition synthetic biology' returns 283 results, versus 'Design synthetic biology' which returns 2,710,000 results. Also, searching for 'Implementation synthetic biology' returns 351,000 results, versus 'Construct synthetic biology' which returns 1,280,000 results.

To summarise, not only were all the *TopLevel* terms in use by SynBioHub-Lab replaced with more colloquial wording, but the nomenclature for predicates between these classes were also changed. Objects are also referenced more directly as what they are rather than using a general term like component. These changes make it easier for experimentalists to understand how to use the interface and exploit its strengths.

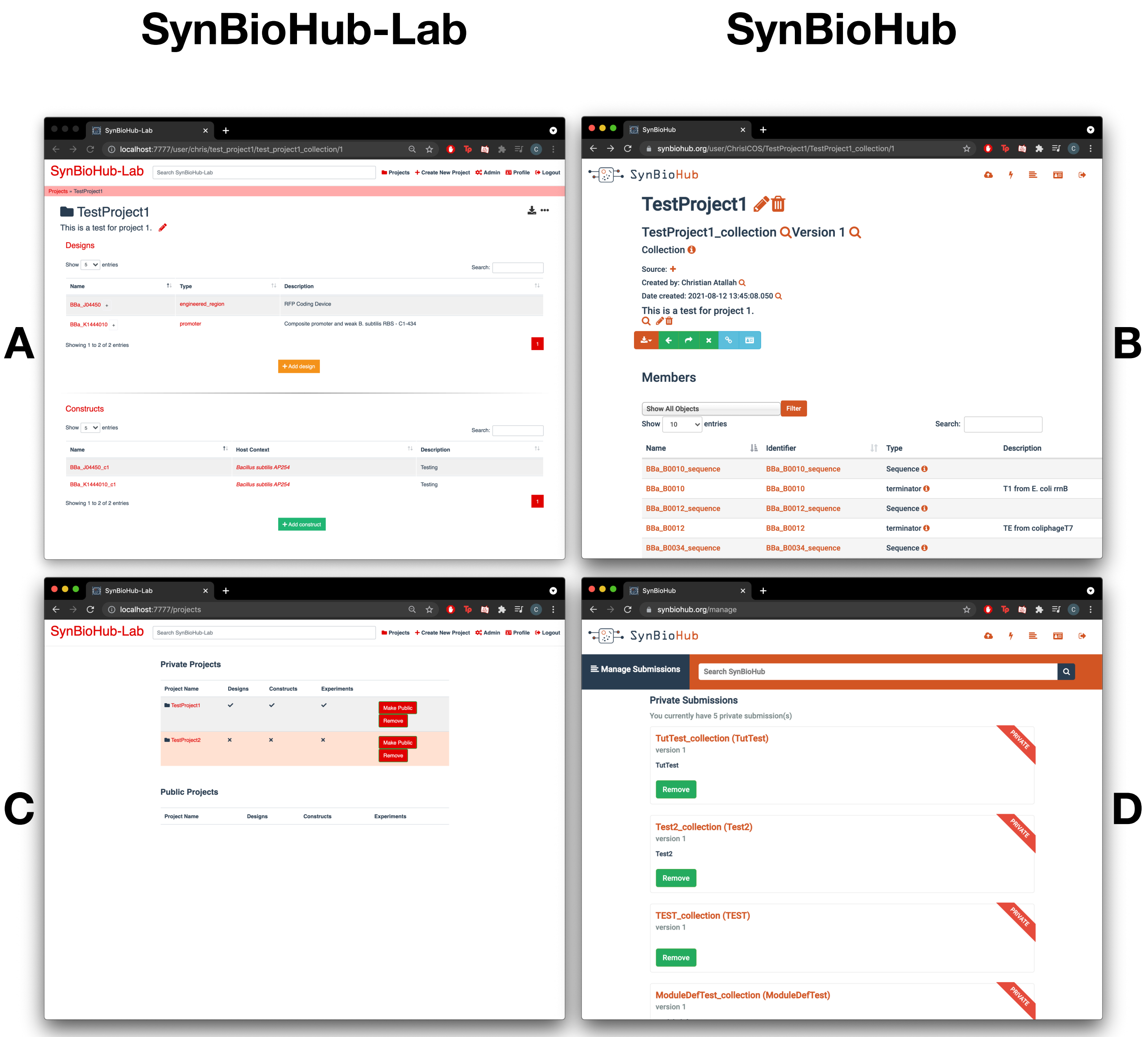
6.3.3 Front-end redesign

The steps in which a user interacts with the front-end, or the flow of control, was redesigned significantly for SynBioHub-lab in a way that makes it more friendly for biologists to use. As mentioned in section 6.3.2, *Collections* were replaced by *Projects* as the starting point of a user. Also, on SynBioHub, a collection must first be created, and requires the uploading of an SBOL-compliant file containing any number of SBOL objects. In SynBioHub-Lab, however, it is first necessary to create an empty project. This change allows users to create and organise projects without necessarily having the necessary SBOL objects ready.

Another major difference between *Collections* and *Projects* is that *Collections* are presented as lists of SBOL items, with no separation between the different types of the synthetic biology lifecycle (Figure 6.16-B). However, in SynBioHub-Lab, entries contained in *Projects* are separated into three major categories: *Designs*, *Constructs*, and *Experiments*. Each of these categories are clearly marked and colour-coded to be recognisable, with submit buttons for *Designs* in orange, *Constructs* in green, and *Experiments* in blue (Figure 6.16-A). Also, each of these three sections of a project contains a bespoke table displaying entries relevant to each part of the cycle; the design table contains SBOL entities representing sequence designs and their hierarchies (*ComponentDefinition* and *ModuleDefinition* objects), while the constructs and experiments tables will contain entries for *Implementation* and *Experiment* entities. These tables subsequently use the nomenclature outlined in section 6.3.2. This separation makes it easier to organise the different three steps of the lifecycle along with the data they produce.

The next major front-end change is in the way lists of projects are displayed. Whereas SynBioHub displays them as simple vertical lists, one for private and one for public access (Figure 6.16-D), SynBioHub-Lab organises them into tables. These tables have four columns, with the first one being the project name, and the final three being one of the three major entities of the lifecycle; designs, constructs, and experiments (Figure 6.16-C). A tick is drawn for each entity type that is present in some project. This change is to help users immediately identify the stage of the synthetic biology lifecycle a project has reached, and helps in distinguishing them by more than just their name.

Finally, the last major difference to the flow of control between the two interfaces is in the addition of entities relating to test and build to a project. On SynBioHub,



TestProject1

This is a test for project 1.

Designs

Show 5 entries Search:

Name	Type	Description
BBa_J04450 +	engineered_region	RFP Coding Device
BBa_K1444010 +	promoter	Composite promoter and weak B. subtilis RBS - C1-434

Showing 1 to 2 of 2 entries

Constructs

Show 5 entries Search:

Name	Host Context	Description
BBa_J04450_c1	*Bacillus subtilis AP254*	Testing
BBa_K1444010_c1	*Bacillus subtilis AP254*	Testing

Showing 1 to 2 of 2 entries

TestProject1

TestProject1_collection Version 1

Collection

Source:

Created by: Christian Atallah

Date created: 2021-08-12 13:45:08.050

This is a test for project 1.

Members

Show 10 entries Search:

Name	Identifier	Type	Description
BBa_B0010_sequence	BBa_B0010_sequence	Sequence	
BBa_B0010	BBa_B0010	terminator	T1 from E. coli rrnB
BBa_B0012_sequence	BBa_B0012_sequence	Sequence	
BBa_B0012	BBa_B0012	terminator	TE from coliphageT7
BBa_B0034_sequence	BBa_B0034_sequence	Sequence	

Private Projects

Project Name	Designs	Constructs	Experiments	
TestProject1	✓	✓	✓	Make Public Remove
TestProject2	×	×	×	Make Public Remove

Public Projects

Project Name	Designs	Constructs	Experiments

Private Submissions

You currently have 5 private submission(s)

TutTest_collection (TutTest)
version 1
TutTest
Remove

Test2_collection (Test2)
version 1
Test2
Remove

TEST_collection (TEST)
version 1
Remove

ModuleDefTest_collection (ModuleDefTest)
version 1

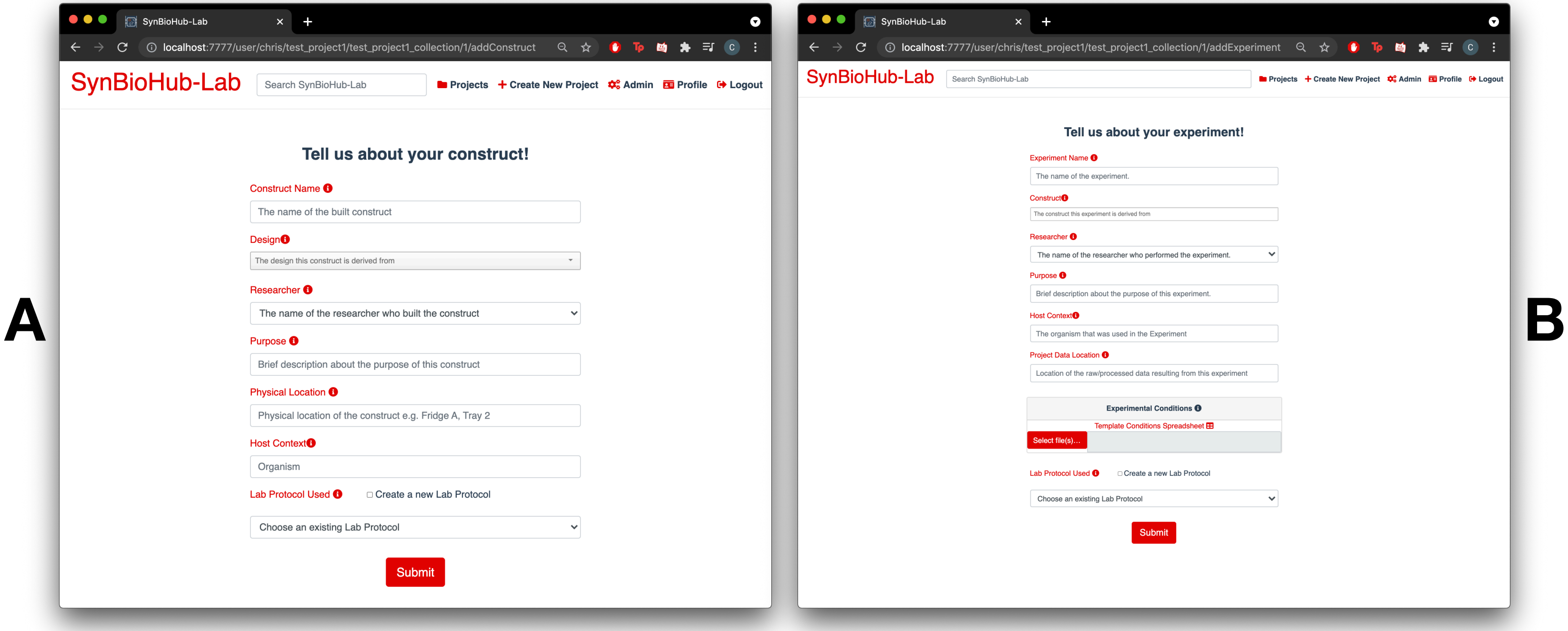
Figure 6.16: Screenshots comparing the front-end of SynBioHub (B, D) and SynBioHub-Lab (A, C). In the latter, there is a more present emphasis on the distinction between the design, build, and test stages of the synthetic biology lifecycle. This is shown in both the collection page (A and B) projects page (C and D).

the only way to currently upload such files is to submit further SBOL files containing *Implementation* or *Experiment* objects. While there are many tools for producing SBOL-compliant files for the design stage such as SBOLDesigner [120], there is a lack of biologist-friendly tools for producing SBOL for the build and test stages. This fact limits users to producing such SBOL data with one of the many existing SBOL libraries like sboljs [124], which is inaccessible to experimentalists.

The solution to this problem presented in SynBioHub-Lab is the addition of form pages that allow for the creation of individual construct and experiments entries (Figure 6.17). These forms are accessed from within a project page by clicking one of the respective submit buttons, and when submitted, automatically generates compliant SBOL in the back-end. These forms also store the provenance of a construct relative to its design (Figure 6.17-A), raw data location and experimental conditions for experiments (Figure 6.17-B), lab protocols for both constructs and experiments, and metadata relevant to build and test, such as the researchers who undertook the processes. This feature makes SynBioHub-Lab one of the first accessible platforms for generating and storing SBOL data about the build and test stage.

6.4 Discussion and Conclusions

While the approaches introduced in the previous three chapters share the purpose of optimising the diversity in an enzyme family panel, the computational methods that help select such panels are not well integrated with the experimental processes that characterise them. There is a lack of tooling that makes the transition from each step of the selection and testing process more complicated; from mining, to analysis, to panel selection, to enzyme characterisation. Furthermore, this lack of tooling limits the iterative potential of the workflow, as there is no concrete way of learning from previous assays for future ones. Finally, there are no suitable repositories for data and metadata produced by each step of this workflow, the storage of which is particularly important for the experimental stage. Therefore, tools that alleviate these gaps were developed to help make the process of exploring the diversity of enzyme families more smooth, especially in the transition from *in silico* to *in vitro* steps.


A
B
SynBioHub-Lab
localhost:7777/user/chris/test_project1/test_project1_collection/1/addConstruct
localhost:7777/user/chris/test_project1/test_project1_collection/1/addExperiment
SynBioHub-Lab
Search SynBioHub-Lab
Projects
+ Create New Project
Admin
Profile
Logout
Submit
Select file(s)...

Tell us about your construct!

Construct Name
The name of the built construct

Design
The design this construct is derived from

Researcher
The name of the researcher who built the construct

Purpose
Brief description about the purpose of this construct

Physical Location
Physical location of the construct e.g. Fridge A, Tray 2

Host Context
Organism

Lab Protocol Used □ Create a new Lab Protocol
Choose an existing Lab Protocol

Tell us about your experiment!

Experiment Name
The name of the experiment.

Construct
The construct this experiment is derived from

Researcher
The name of the researcher who performed the experiment.

Purpose
Brief description about the purpose of this experiment.

Host Context
The organism that was used in the Experiment

Project Data Location
Location of the raw/processed data resulting from this experiment

Experimental Conditions
Template Conditions Spreadsheet

Lab Protocol Used □ Create a new Lab Protocol
Choose an existing Lab Protocol

Figure 6.17: Screenshots of SynBioHub-Lab's novel forms for adding data about the build (A) and test (B) stages of the synthetic biology lifecycle.

The first tool developed, called IntEnz-Lab, is a bespoke platform that integrates the mining and panel selection process and the enzyme characterisation data procured for said panels. This integration helps promote an iterative approach to the functional profiling of enzyme families. The second tool developed, called SynBioHub-Lab, is a repository for storing data and metadata resulting from the synthetic biology lifecycle. As a biologist-friendly rewrite of SynBioHub with novel functionalities, SynBioHub-Lab facilitates the storing of data and metadata produced by characterisation assays, which will pass through every step of the lifecycle.

6.4.1 Strengths and limitations

IntEnz-Lab is a novel web-interface that integrates two important command-line tools used by the workflow shown in Figure 6.2 - *hmmsearch* and the MDP algorithm (Chapter 4) - with the intention of becoming a central hub for enzyme family dataset building and panel preparation from metagenomes. It is also meant to be a central repository for storing the results of such operations on enzyme families of interest, due to a database schema and bespoke API that makes individual families the central entity of the interface to which other data are integrated. Importantly, this integration includes characterisation data for sampled panels, and is done in such a way that future panels can be guided through functionality that automatically masks tested enzymes. This centralisation of relevant tools, their metagenomic inputs, the resulting panel and profiling outputs, and is an important feature and strength of IntEnz-Lab.

However, IntEnz-Lab is missing some functionalities for it to reach its potential as a central hub for enzyme characterisation research. For example, there is currently only one method of sequence masking - based on testing status - which limits the different ways a can iterate for the next enzyme panel. Furthermore, panel preparation without performing an analysis of the selected MDP panel is not recommended, as discussed in section 4.4.2, but IntEnz-Lab does not yet have the functionality for such analysis to be performed on its interface. Finally, the current way of uploading characterisation data through a template spreadsheet requires a user to be consistent with the naming of enzyme entries, and chemicals and their ChEBI IDs, which could cause issues like duplication of data and incorrect data integration.

A major strength of SynBioHub-Lab is the reworking of the vocabulary used by the interface, through the replacement of SBOL terms with a more biologist-friendly lexicon. This change in terminology, along with a new flow of control that emphasises the separations between design, build, and test, makes the interface significantly more suited to experimentalists, including those carrying out enzyme characterisation studies. Also, SynBioHub-Lab is one of the few tools that can be used to represent synthetic biology entities of the build and test stage in an SBOL-compliant way, thanks to novel forms that allow a user to create them.

However, an important limitation of the current version of SynBioHub-Lab is that it is locked to version 2.3 of the SBOL specification, the latest version of which is 3.0. Such a drawback is likely to result in SynBioHub-Lab being incompatible with newly generated SBOL data and implemented tools. Also, an inconvenient aspect of the current SynBioHub-Lab flow of control is the inability to add entries of any type in batches, as currently each must be uploaded or created individually. Finally, while both SynBioHub-Lab and IntEnz-Lab help plug in gaps in tooling that are necessary for an iterative approach to exploring enzyme families, there is no integration between them, reducing the potential of the approach.

6.4.2 Future work

Most of the future work for both IntEnz-Lab and SynBioHub-Lab lies in the development of further functionality that can make each platform more useful. For IntEnz-Lab, the addition of more sequence masking options would greatly improve the iterative aspect of the platform. Also, with more masking options, one could also allow for multiple such options to be applied in the same query. Some new useful masking queries that could be implemented include:

- Discarding sequences from panel selection that have been tested with a specific list of chemical compounds, rather than any compound. This masking option could be useful if sequences with desired activity profiles have been identified for some tested substrates, but not all, allowing to refine the search to a more specific sequence-function space.

- Discarding sequences from panel selection that have some user-specified sequence identity threshold with an already tested enzyme. While this query would require more work as IntEnz-Lab does not store sequence identity, a graph database like the one in use in this tool is perfectly suited for implementing this extra functionality. Such a masking option would allow for reducing the sequence space further by also discarding sequences with high levels of identity to characterised enzymes. Doing so could save time spent on needing to analyse and refine a panel with visualisation methods.

Another high-impact avenue for future work on IntEnz-Lab would be to add visualisation functionality to the interface. Specifically, if SSNs, CSNs, and trees could be generated for individual enzyme families, along with the annotation and highlighting of tested enzymes, it would make the analysis step of panel preparation more accessible. Finally, integrating the uploading of characterisation data into the interface in a more dynamic way instead of depending on an spreadsheet would lead to less user errors, as the interface would be able to validate any new annotation.

As for SynBioHub-Lab, the most impactful change to be made to it lies in the refactoring of the codebase to use SBOL3, which is the most recent major version. Such refactoring would likely be time-consuming, but it would allow for a healthier back-end infrastructure as SBOL3 lowered the complexity of the model significantly. This change would also allow SynBioHub-Lab to be more compatible with other state-of-the-art SBOL tools and the data they produce.

On an interface level, another major change to SynBioHub-Lab that would be beneficial is added functionality for the uploading or creation of entries in batches. Finally, integrating both IntEnz-Lab and SynBioHub-Lab into a single toolkit where both platforms can communicate with each other would be highly impactful. For example, it could allow for seamless links from tested enzymes on IntEnz-Lab to their respective entries on SynBioHub-Lab to identify which organisms they were transformed into at build and test, or to locate raw data associated with them in more detail.

6.4.3 Conclusions

In this work, two novel platforms were developed to help tackle gaps in tooling to increase the level of integration between the *in silico* and *in vitro* side of enzyme family exploration. IntEnz-Lab was implemented as a web-interface that firstly unifies the dataset building and panel selection processes, and secondly integrates them with the enzyme characterisation made possible by them in such a way that gained knowledge can be iterated and improved upon. SynBioHub-Lab was then designed as a repository for metadata resulting from the *in silico* design step and the *in vitro* build and test steps of the synthetic biology lifecycle. Given the importance of data and metadata like plasmid designs and experimental protocols used, having a bespoke storage platform for them that uses the well-recognised standard of SBOL was a crucial gap, which is now filled by SynBioHub-Lab. Both of these tools help promote an iterative approach to the exploration of enzyme families that is more successful over the long term.

7

Discussion and conclusions

Contents

7.1 Introduction

The use of enzymes as biocatalysts in industry has many varied benefits when compared to classic chemical synthesis methods, such as higher stereoselectivity [17] and the simplification of pathways [18], while being more environmentally-friendly, often with higher yields [19].

The application of biocatalysts depends on the selection of enzyme panels that catalyse varied chemical transformations of interest, often for specific enzyme families. These enzymes are often tested in panels, to ensure that the best match for a process is identified. However, the construction and curation of panels is currently limited by the ever-increasing amount of uncharacterised enzymes in public databases [27]. Therefore, efforts to bridge knowledge gaps in enzyme families through laboratory-based characterisation of panels is necessary for expanding the catalogue of biocatalysts useful to industry.

However, as was discussed in section 2.2.3 of the Background, current methods for the generation and characterisation of diverse enzyme panels have many limitations, including an over-reliance on sequence identity, enzyme sampling methods that are un-optimised, and a tooling gap for iterating on characterisation studies. Therefore, novel methods that help optimise this process of panel curation have high value, due to a shortening of the time and burden of knowledge necessary to undertake mass characterisation projects of enzyme families, and improved tooling. To this end, two principal aims were identified:

1. The development of new computational methods for building diverse sequence panels from enzyme families
2. The building of tools that promote an integrated and iterative framework for the characterisation of enzyme families in the laboratory

This final chapter restates the primary approaches used to tackle these aims, identifies the research gaps that were filled by them, and frames the research in the context of already existing work. The chapter is divided into two main sections - one for each aim. Finally, a unified framework for the iterative and integrative characterisation of diverse enzyme families based on this thesis is proposed, along with a discussion on the research gaps remaining for future work.

7.2 The generation of diverse enzyme panels

The catalytically diverse selection of enzymes from an enzyme family dataset for the experimental profiling of putative biocatalysts was the key theme of this research project. Indeed, the improvement of the panel generation process was a direct aim of the research described in three of the four research chapters (Chapters 3, 4, 5). This section summarises how each chapter contributes to this overarching aim.

7.2.1 Functional analysis of enzyme families

As was discussed in section 2.2.2, the current methods of sampling from larger enzyme datasets to select diverse enzyme panels include the functional analyses of enzymes to make informed decisions on panel selection [27, 34]. However, as was discussed in section 2.2.3, these analyses are often limited by a a need for prior knowledge about an enzyme family of interest, reducing accessibility. There are other approaches, such as phylogenetic analysis, which are useful at delineating the diversity of an enzyme family without much prior knowledge. However, these methods are based on sequence identity methods that are not as applicable to enzymes due to more stringent evidence requirements for annotation transfer [44].

Chapter 3 introduced a novel approach for the functional analysis of enzyme families that tackles the limitation of sequence identity methods. CSNs, or coevolution similarity networks, are similarity networks that visualise the delineation of enzyme families based on patterns of coevolving residues. Coevolving residues have long been used as proxies for residues in contact [137], and patterns of them therefore contain information inherent to the tertiary structure of enzymes. Also, coevolving residues are often important to enzyme function [141–143]. Therefore, CSNs were developed with the aim of better grouping enzymes based on enzyme features more specific to function than raw sequence identity methods. The results of this chapter in section

3.3 showed that CSNs are capable of more sensitive functional groupings at the edge of the sequence identity twilight zone in four different enzyme families.

Recent developments in the functional analysis of enzyme families have similarly developed methods less dependent on sequence identity by also including other enzyme features. For example, Holliday and coworkers combined sequence similarity with structural and chemical similarity in an analysis of three enzyme superfamilies [200]. Approaches that similarly visualise structural and chemical similarities are useful for diverse enzyme panel creation as they provide a more detailed view of an enzyme family, which can help inform the selection process. CSNs, due to the inherent structural information present in residue-residue coevolution data, are able to represent similar structure-based relationships, and therefore offer similarly useful views of enzyme families. Indeed, many CSN linkages were made between enzymes of low sequence identity, but high tertiary structure similarity. Importantly CSNs were able to capture structure relationships without requiring resolved tertiary structures, as CSNs are built from primary sequence alone. This distinction sets CSNs apart as being more accessible than approaches similar to Holliday and coworkers' [200], which explicitly require PDB files to construct structure similarity networks.

While enzyme function does not consistently transfer based on arbitrary sequence identity thresholds [44, 200], such arbitrary thresholds are still used to automatically annotate enzymes on public databases, with negative repercussions [133]. As recently as 2021, over 78% of sequences representative of the S-2-hydroxyacid oxidases were experimentally found to be misannotated [201]. Erroneous functional annotations on public databases directly impacts panel selection by giving an incorrect view of the diversity of an enzyme family, from which panels are sampled. However, as CSNs were shown to computationally highlight misannotations in the crotonase family, CSNs offer a method of identifying such misannotations *in silico*, rather than *in vitro*. The advent of approaches like CSNs can therefore play a role in fixing some of the negative repercussions of sequence identity-based family analyses.

To conclude, CSNs can help perform *in silico* analyses of the catalytic diversity of enzyme families. As enzyme panel selection is currently based on such analyses [178], CSNs can therefore be used as a direct refinement tool in panel selection owing to more sensitive functional linkages.

7.2.2 Automatic selection of diverse enzyme panels

CSNs, much like SSNs and phylogenetic trees, currently require manual interpretation and iteration of parameters like similarity thresholds to optimally analyse enzyme families. This process can be time-consuming and difficult, especially without an already rich understanding of the enzyme family at hand. There was therefore a need for a complementary method of automatically sampling highly diverse subsets of enzymes from a larger dataset, which could then be optimised using CSNs and similar methods.

In Chapter 4, heuristic algorithms were developed for solving the maximum diversity problem (MDP) on enzyme datasets. The hypothesis explored in this chapter was that MDP-solved subsets would automatically generate subsets of enzymes with high catalytic diversity. Indeed, this approach was shown to produce, without prior knowledge or parameter optimisation, representative subsets that are diverse in catalytic profiles on three different enzyme families using sequence identity matrices as input. Sampled subsets were shown to well represent the delineation of functional diversity using sequence identity, while also correctly summarising the clustering of enzyme families using the coevolution similarity metric introduced in Chapter 3.

As far as the author is aware, there does not yet exist study on the automatic and optimised selection of diverse enzyme panels from larger sets of enzymes. Recent developments sample diverse panels by computing clusters of similar enzymes based on factors of varying complexity, from which enzymes are then manually sampled from. For example, Vanaceck and coworkers [34] manually sampled a panel of 20 enzymes based on a balance of taxonomy, solubility predictions, predicted active site volume, and other factors. While this approach was successful, as the panel was proven to be diverse, it is a highly involved and manual process that also requires the computation of various data. MDP-based panels instead offer a novel metaheuristic approach that also selects highly diverse panels, but without the time and analysis complexity of an approach like Vanaceck and coworkers'.

A simpler approach is the one used by Velikogne and coworkers [32]. Velikogne and coworkers produced a phylogenetic tree out of a larger set of enzymes, and created their panel by randomly sampling enzymes from the main sub-branches of the tree. While simpler and more efficient than Vanaceck and coworkers [34], sampling from sub-branches of a tree still requires manual interpretation to delineate the clade boundaries that are sampled from. In comparison, MDP-based panels do not have such requirements; the only user input required is the panel size, which is a parameter that does not need to be optimised, as its value is likely to depend mostly on laboratory capacity.

While enzyme panels generated using the MDP approach were shown to be catalytically diverse, it was also clear that some oversampling of certain sequence space naturally occurs using this method. MDP panels would therefore benefit from manual interpretation and refinement, similar to the ones performed by Vanaceck and coworkers [34] and Velikogne and coworkers [32]. Novel methods introduced in this research like CSNs could similarly be used. In the context of refinement, an MDP provides a diverse initial subset, which simplifies the analysis and interpretation of applied CSNs.

In conclusion, the MDP-based approach helps circumvent the requirement for burden of knowledge in diverse panel generation. MDP subsets also reduce the complexity of analysis needed to interpret structures CSNs, by providing a starting point panel that is already likely to be diverse.

7.2.3 Neural network-based de novo enzyme generation

Chapter 5 describes a generative approach for increasing the amount of known diversity in an enzyme panel by generating it *de novo*. Using a neural networks approach, a discriminant autoencoder was trained on all of the enzyme sequences available on Swiss-Prot, with the goal of learning an inherent enzyme model. Then, this learned space was sampled from using a template set of aldo-keto reductase (AKR) enzymes to generate novel and synthetic candidates of that family. These synthetic aldo-keto reductases were pushed through a semi-automated filtering pipeline to keep high quality candidates based on user-defined constraints.

The filtered synthetic sequences were assessed using both *in silico* and *in vitro* methods. Specifically, these putative synthetic enzymes were used as input to bioinformatics tools like I-TASSER [82] and DEEPre [28] to judge their potential viability, with promising results. In the laboratory, they were shown to be capable of overexpressing in *E. coli*, while being mostly insoluble, likely due to misfolding. While the method therefore does warrant further improvements, some of which are outlined in section 5.4.3, it was shown to have promise as an approach for artificially increasing the diversity of enzyme panels through the generation of synthetic but viable candidates.

There have been many other recent approaches to the *de novo* generation of functionally viable enzymes. These approaches include other neural networks-based methods, such as the work of Repecka and coworkers [178]. Using generative adversarial networks (GANs), Repecka and coworkers generated 55 synthetic malate dehydrogenase candidates, of which 13 showed activity. Russ and colleagues [202] successfully produced five artificial chorismate mutases through the sampling of the evolutionary history of the enzyme class. Furakawa and coworkers [203] used a state-of-the-art phylogenetic technique called ancestral sequence reconstruction to successfully generate two ancestral enzymes of 3-isopropylmalate dehydrogenase.

The artificial generation of functioning enzymes is therefore quickly becoming a heavily-researched area. The research undergone in Chapter 5 does also make a unique contribution to this rising field. Specifically, while the aforementioned three approaches were all bespoke to the enzyme class level, the method explored in this research was generalised to the enzyme family level. The discriminant autoencoder architecture was trained on a dataset of hundreds of thousands of enzymes, and generated synthetic enzymes displaying a high degree of similarity to a specific family. This similarity included a high overall sequence identity to canonical enzymes, the presence of domains and other sequence signatures, and the prediction of functional and structural annotations that are consistent with those of the given enzyme family. Therefore, in spite of negative experimental results, the work undertaken here can function as a platform for future work that is able to generate panel diversity on an enzyme family level.

7.2.4 Conclusion

All three strands of work therefore contributed to the aim of optimising the panel generation process. With a novel functional analysis and visualisation method in

CSNs (Chapter 3), a heuristic algorithm for automatically generating catalytically diverse subsets of enzymes (Chapter 4), and a promising approach for the synthetic generation of novel enzymes (Chapter 5), this thesis made significant contributions in the plugging of specific research gaps for the generation of diverse enzyme panels.

7.3 Tooling for an integrative and iterative framework of enzyme family characterisation in the lab

The initial three research chapters expanded specific aspects of enzyme panel generation, and while they can theoretically be integrated together in a framework as described in sections 3.4.2, 4.4.2, and 5.4.2, no tooling yet achieves this integration. Furthermore, as described in section 2.5.3, there are significant tooling gaps in the representation of the design-build-test-learn (DBTL) cycle in SBOL. These two gaps limit the potential of an integrative and iterative framework for the characterisation of enzyme families in the laboratory. Consequently, the second aim of this thesis tackled these limitations. In this section, the ways in which the two tools discussed in Chapter 6 help counter these limitations are recalled.

IntEnz-Lab is a platform that integrates some of the *in silico* techniques for enzyme panel generation utilised in this thesis, including the mining of hits from metagenomes in a way that is enzyme family-focused, and the automatic generation of diverse panels from hits using the MDP-based methods of Chapter 4. Also, IntEnz-Lab provides the novel functionality of integrating enzyme characterisation data produced in the laboratory for the purposes of guiding future rounds of panel selection, thereby making the process iterative .

SynBioHub-Lab is an SBOL-powered repository for whole lifecycle synthetic biology experiments, including enzyme characterisation. As a rework of SynBioHub [109], SynBioHub-Lab allows for the storage of synthetic biology designs, while also providing functionality for the storage of metadata about constructs and experiments. This novel functionality allows a user to take advantage of SBOL's powerful modelling of the build and test stage of the lifecycle, making the use of the DBTL cycle for synthetic biology and enzyme characterisation studies accessible for the first time.

Both of these tools help achieve the second central aim of this thesis, by providing accessible and simple tooling that not only helps unify the novel methods of the thesis, but also allows for a better exploitation of the benefits of the synthetic biology lifecycle.

7.4 An applied framework for lab-based exploration of enzyme family diversity

While IntEnz-Lab and SynBioHub-Lab have limitations in their current iterations, their various strengths help formulate an applied framework for lab-based exploring of enzyme family diversity. This framework uses the two tools introduced in this chapter, but also the *in silico* methods discussed in Chapters 3, 4, and 5, along with other established methods like *hmmsearch*-based dataset building and SSN construction. Also, this framework helps fulfil the four different stages of the synthetic biology lifecycle; design, build, test, learn.

This framework is described with using a curated HMM for an enzyme family of interest and multiple sets of translated open reading frames from different metagenomes as a starting point (Figure 7.1).

First, IntEnzLab is used to upload the different metagenome files. Then, the enzyme family of interest is created on IntEnz-Lab using the existing profile HMM as input. Next, each metagenome is mined using *hmmsearch* on IntEnz-Lab for potential sequences of that family. The resulting hits is then used as input for the MDP solver to sample a diverse panel. Then, SSNs and CSNs are generated for the hits. The aforementioned MDP panel is then refined using these similarity networks. Then, another MDP panel is produced to act as template for the generation of synthetic sequences. After filtering of unviable sequences, a panel of synthetic sequences is selected.

The natural and synthetic sequences are then combined into a single larger panel. The sequences of this combined panel are then codon-optimised into DNA sequence, and then inserted into relevant plasmid designs. These designs are finally uploaded to a new SynBioHub-Lab *Collection*. This step and preceding ones fulfil the **design stage**. Once a design from the panel is built in the laboratory, a construct is then also added to

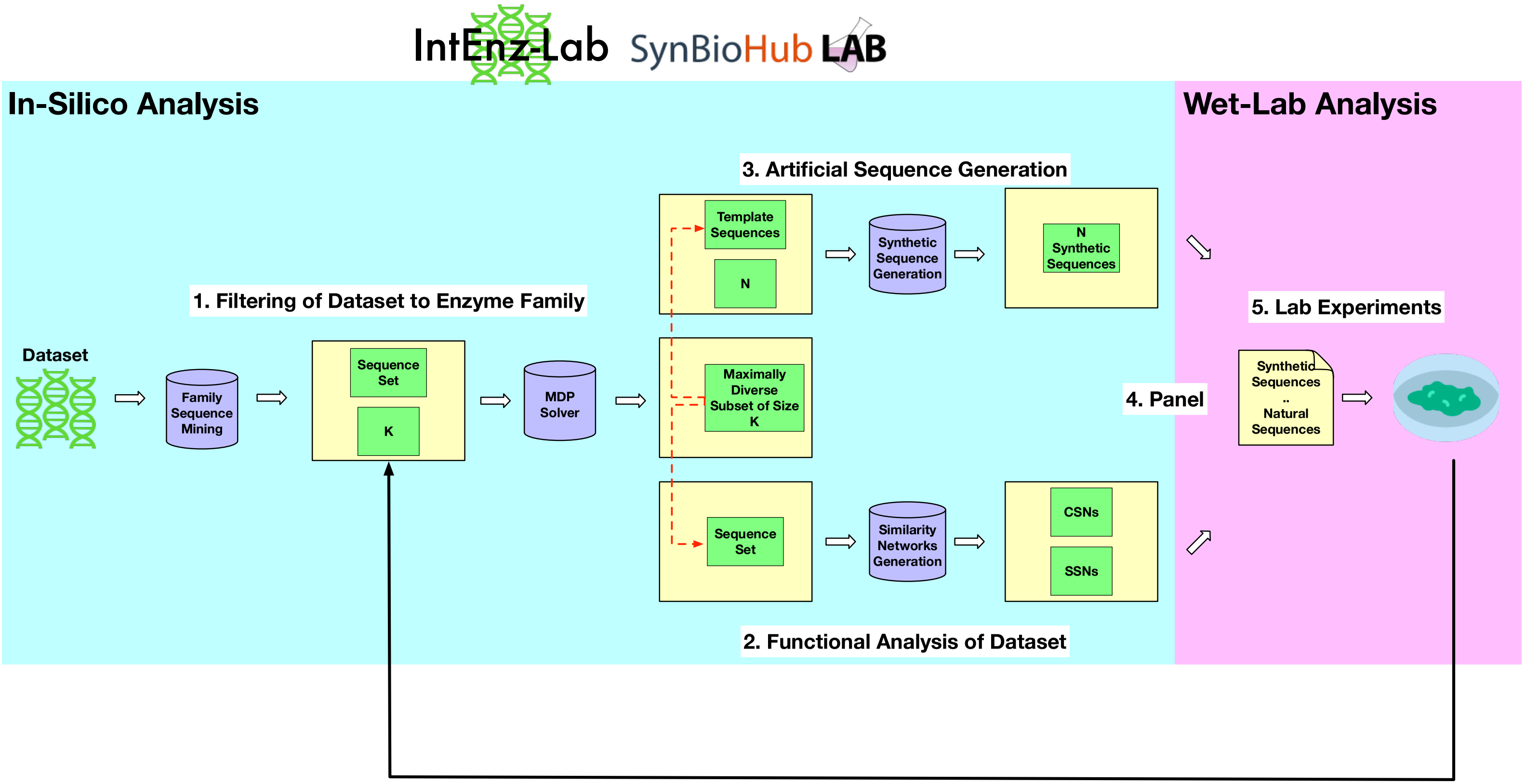


Figure 7.1: Diagram describing the integrative and iterative framework for the lab-based exploration of enzyme family diversity that uses previously existing approaches on top of the new methods introduced in this thesis.

SynBioHub-Lab, along with any validation and quality assurance measures. This step fulfils the **build stage**. Once the panels are characterised, SynBioHub-Lab can then be used to create entries storing relevant metadata and experimental conditions. The characterisation data itself can also be uploaded to IntEnz-Lab for integration with its knowledgebase. This step fulfils the **test stage**. Finally, the next iteration of panel generation is guided by IntEnz-Lab using its native sequence masking functionality. This step fulfils the **learn stage**.

This workflow, starting with a large set of metagenomic sequences and a profile-HMM for an enzyme family of interest, is able to utilise the methods discussed in this research to iteratively generate and characterise diverse enzyme panels. The integration of *in silico* methods and experimental data thus enables the semi-automatic exploration of an enzyme family's diversity over multiple iterations. With the use of tools like IntEnz-Lab and SynBioHub-Lab, the framework is made more accessible, along with key functionalities that make the DBTL-based functional analysis of enzyme families a reality.

However, while this workflow can currently be performed, future work is needed to simplify and optimise it further. First of all, the discriminant autoencoder described in Chapter 5 needs to be improved further before synthetic sequences it generates are worth adding to panels, as was discussed in section 5.4.3. Also, the addition of further sequence masking options to IntEnz-Lab would significantly improve the learn stage, with some examples given in section 6.4.2. Finally, it would make the framework more accessible if the analysis of sets and subsets created on IntEnz-Lab could be performed in-place, which would require the further integration of tools, including similarity network generation.

7.5 Conclusion

With the power of similarity networks like CSNs, which can be used to help refine automatically-selected panels produced by the MDP method, and the added diversity of synthetic sequences generated by an improved autoencoder, the panel curation process has been optimised by the novel approaches introduced in this thesis. Specifically, these methods successfully tackle three important gaps in current panel preparation approaches - CSNs address the dependence on sequence identity for assessing functional groupings, MDP sampling addresses both the requirement for prior knowledge for diversity assessment in enzyme families and the manual interpretation that is often necessary to perform such assessments. The discriminant autoencoder explores an avenue for automatically generating novel diversity in an enzyme family.

Also, novel tools like IntEnz-Lab and SynBioHub-Lab help integrate the *in silico* and *in vitro* processes of enzyme selection and characterisation. The integrative and iterative framework that is born from the combination of already existing approaches and the ones introduced in this thesis is one that can significantly optimise the thorough analysis of enzyme families. To conclude, the work contained in this thesis is a significant step towards the important goal of creating a more diverse portfolio of biocatalysts, which is necessary for making the substantial benefits of enzymes more accessible and useful to industry.

A

Enzyme family datasets

Multiple different datasets of enzyme families were built from public databases in this thesis. Specifically, Pfam IDs representing specific families were used to retrieve entries from Swiss-Prot. Here, the different lists of UniProt accessions making up each dataset is written.

Transaminase class I&II (PF00202), referred to as Trans241 in Chapter 3:

```
P22256, P12995, P9WQ81, Q9I700, P40732, P50457, P53555, P18335,
H8WR05, O30508, P77581, P23893, P42588, P24630, Q93R93, P9WPZ7,
B0VH76, Q9APM5, Q7M181, Q53U08, O52250, P59324, Q88RB9, P22805,
P38021, M1GRN3, Q5SHH5, P16932, Q8X4S6, P59317, Q8Z1Z3, Q9I6J2,
O66442, P59321, P28269, P46395, P57600, Q7N9E5, P59086, P53656,
P36568, P94427, Q01767, Q55665, Q6L741, Q9I6M4, Q9X2A5, E1V7V7,
P12677, P0A4X7, O25627, O66557, Q74CT9, P45488, P44426, Q2FVJ6,
Q9FCC2, P9WQ79, B7GHM5, Q9ZKM5, P57379, Q89AK4, Q8K9P0, P9WQ80,
Q31QJ2, A0QYS9, P40829, P9WMN9, P63505, Q8DLK8, Q9RW75, Q8D0D7,
Q8YS26, A2BVE5, D2D3B2, P9WQ78, B1X023, Q3AWP4, Q7NN66, Q818W2,
Q8ZPV2, P56744, Q9PIR7, Q4H4F5, Q7NPI4, A9WIS7, Q8KAQ7, B0JPW6,
B2J7M9, Q81M98, Q7VDA1, A2BPW6, B3EI07, B4S3Q6, Q2JS70, A3PBK8,
B3QSA6, Q2JMP7, A5GMT2, Q110Z9, A1BJG8, B7KA18, B4SGW1, B8G822,
B3EKJ7, A2SKQ7, A2C0U2, A9BEA5, A7NKV1, Q3M3B9, Q7V2J3, A5UU40,
Q3AP59, Q3B1A1, B8HYK1, B0CC57, A8G3J9, Q31C50, B3QRD2, A4SGT2,
Q7VMS5, Q7VAS9, Q8CSG1, B1XIT5, B0TFV0, B7K2I1, Q7V677, Q7U598,
Q0I8G1, A5GUJ2, Q3ALU9, Q7V0G0, P9WPZ6, Q46GT9, A2C7I7, P36839,
Q7MAE6, B9LKS0, Q89VE9, Q7W7H6, Q885K0, Q89LG2, Q8Y6U4, Q9L1A4,
P59320, Q8EHC8, Q9KU97, Q9AAL3, Q9KNW2, P73133, Q8P5Q4, P24087,
Q9ZEU7, Q8X4V5, Q8FL16, P30949, Q725I1, Q9KLC2, P30900, Q9RWW0,
Q9AP34, A0QR33, B7GIK0, B7GH35, Q2G283, A8ALD5, Q5ZVA6, Q2S1S3,
A7MUU9, Q4L7G9, A4FPX3, Q7MHY9, Q1IWZ8, A5WC94, P0C2D9, B9DZG0,
B2SKS0, A3NCF3, Q9CC12, Q9JYY4, P59316, Q5H3I5, Q8Y6J9, Q3JPN1,
O66998, A3MMQ8, P46716, Q882K8, Q9CHD3, A1V1L0, P59322, Q8UI71,
P59315, Q9K8V5, P63566, Q99T15, Q2FXR4, Q7VTJ7, Q9A652, Q97GH9,
Q59282, O08321, Q98BB7, Q59928, Q8XWN8, P59319, Q87L20, Q7WKW5,
Q8CUM9, Q9KEB0, Q9JRW9, Q8CRW7, Q9K8G3, Q9CNT1, Q828A3, Q7VSH3,
Q8R7C1, Q92BC0, Q92SA0, P54752, Q9PDF2, Q05174, Q92AX5, Q6QUY9,
Q8PH31, Q82UP3, Q8FTN2, Q5HN71, Q7WDN7, P63567, P59323, Q72RH8,
Q7MH19, Q7U5R5, Q5HER0, P63569, Q9JTX9, Q7V8L1, Q5HP24, P48247,
Q7A4T5
```

Short-chain dehydrogenase (PF00106), referred to as SDR142 in Chapter 3:

```
P14697, P39831, P80702, P07914, Q9RA05, Q8KES3, P9WGS9, Q9WXG7,
Q48436, Q8RJB2, Q82IY9, Q9KWN1, P0CI31, Q1R183, P47227, P74167,
P05707, P07772, Q1QU27, P0CI32, P0A9P9, Q93UV4, A0R610, A7B4V1,
D3U1D9, P96825, P16544, Q46381, Q8XA72, P9WGP9, B7NRJ0, Q04520,
B7N6C8, Q83QJ8, P39071, B7M7P4, A7ZPY4, Q3YZ12, C4ZXB6, B1IVT6,
B7LDD3, B1XB16, B6I5B4, Q0T1X8, B1LNJ7, A8A347, B5Z114, Q31XU9,
P23102, P9WGS8, P66784, Q7N4V7, Q8FHD2, P9WGS3, P69936, P69935,
Q8X505, Q83RE8, P72220, P08694, C8WMP0, Q59987, P45375, Q6F7B8,
Q9L9F7, P47230, P08088, P50204, P50206, P37079, P17611, P50203,
P50202, P50201, P27874, Q01198, P39577, E3VWK2, Q5HKG6, E3VWI6,
O66148, P00335, Q9ZAU1, P0A9Q0, P37694, P31808, P43168, O32099,
Q8U8I2, P9WGQ7, P13859, P9WGS1, P16542, P21158, Q9X6U2, P37959,
P41177, P0AFP4, P9WGR5, O05730, P54554, O34782, O32291, P9WGS0,
O32229, P14802, P25145, P66778, Q53877, Q9ZKW1, P9WGS2, P0AFP5,
P9WGP8, P25970, P45200, P9WGR3, Q4L8Y1, Q99RF5, Q2FVD5, Q4A054,
Q49WS9, Q8CN40, Q92EK7, P55434, P9WGR2, P9WGQ6, Q5HD73, P9WGR4,
Q5HLD8, Q2FE21, P66780, Q7A3L9, Q8NUV9, Q6GDV6, Q6G6J1, Q03326,
P35320, O34896, P9WGR7, P44481, P39884, O32185
```

Enoyl-CoA dehydrogenase (PF00106), referred to as Croto99 in Chapter 3:

```
P0ABU0, P9WNP5, A5JTM5, Q8KLK7, Q9XB60, Q5LLW6, Q93TU6, O69762,
P31551, Q9LCU3, P52045, P76082, O85078, P77467, Q5HH38, P23966,
Q39TV7, Q8DR19, Q7CQ56, P52046, Q2LXU2, O87872, A0QRD3, Q84HH6,
Q8DSN0, G4V4T6, P94549, Q8GB17, O87873, Q0AVM1, G4V4T7, P44960,
P40802, Q4L549, Q49WG8, Q8ZRX5, Q5HQC3, P0ABU1, Q99V48, Q8CPQ4,
Q9CLV5, Q7A6A9, Q6GI37, Q8NXA0, Q6GAG7, O32178, Q8XA35, P59395,
P9WNP4, Q8FLA6, P40805, P9WNN9, P9WNP1, P24162, O34893, O07533,
P45361, O07137, P9WNN8, P53526, P9WNN4, A1KN36, Q73VC7, Q50130,
P95279, P64015, Q52995, A0QJH8, P9WNN3, A9MYJ5, Q0TLV3, B5F749,
A9MR28, B4T6J5, Q5PIL1, B1LFW9, C0Q4L2, B4TIG9, B4EY26, B5RGA4,
B4TWR3, A8ALR7, B1IRE0, B5R1Q9, Q57TJ1, B5FHG4, B5BL54, Q8Z9L5,
Q7U004, P64019, Q7TXE1, P9WNN6, P9WNN7, P9WNP0, P64017, P9WNN5,
A5U753, P9WNN2, P31907
```

Transaminase class III (PF00155), referred to as Trans986 in Chapter 3 and ATF in Chapter 4:

```
O66875, O67781, B5Y9Z4, B3QLR6, Q8KDS8, B5ELF7, B7J3Y7, B5YFU5, P21633, P08080, P08262, P18079, P43089, Q92G23, A0L3L7, A5FZN8, A7HP29,
A9HJ57, A9W106, B0SZS9, B0UKC8, B1LYP9, B1Z7Y8, B2IH50, B4RFX5, B7L0L2, B8ERL9, B8GVE2, B8IBW2, Q0APZ9, Q0BUV6, Q2W3L2, Q9A7Z1, Q06191,
A3PMF8, O86459, O87320, P58350, Q02635, Q1RGV0, Q4UND3, Q68XV9, Q92JE7, Q9ZE56, A9H311, P34037, Q0BVW4, Q51687, Q5LNM6, A0KB03, A1ITK1,
A1K6Q1, A1KVF6, A1TTV0, A1V819, A1VUJ6, A2S7R1, A2SD53, A3MNG3, A3N522, A3NQS3, A4G5N9, A4JIB5, A6SU64, A9AE46, A9BV10, A9HVE9, A9M251,
B1JZD9, B1Y500, B1YNS0, B2AG98, B2SWS7, B2UBJ3, B4E9L6, B4RP93, Q0AE73, Q0BBD6, Q0KF88, Q12D74, Q12F38, Q146K3, Q1BT36, Q1GZA7, Q1LS75,
Q2T1Q2, Q2Y9Y8, Q39CE6, Q3JWR6, Q3SLX9, Q477A3, Q47C04, Q5F6R6, Q5NZF5, Q62MX1, Q63Y23, Q7NPW2, Q7WH76, Q81ZZ4, Q8XZC3, Q9K0U0, Q82WA8,
Q6AL81, A1AQT1, A5G6I9, A7HG96, A8ZUS7, B3EAE0, B4UCB1, B5EEV8, B8J3V0, B8J637, B9M8U3, O25320, Q1DCV8, Q2IF62, Q39XE0, Q3A3Z8, Q749W3,
Q9ZLN3, P44425, P72173, Q56114, P71348, P77434, Q9HUI9, P09053, P97084, Q9I468, A7N6R9, A0KIC7, A1JS67, A1U4B1, A1WVM6, A4SPR6, A4TNQ5,
A4VR87, A4W8B8, A4XZR8, A5UCE4, A5UJ44, A5VXF2, A6T6L6, A6UYW1, A6W0Y0, A7FKM8, A7MJ02, A7MX30, A7ZJI5, A7ZY32, A8AJ11, A8GBC6, A9MJE5,
A9MTI7, A9R3C9, B0KJ54, B0RMR1, B0U6J0, B1IXJ2, B1JE54, B1JSS3, B1LM67, B1X7A6, B2FLM5, B2I9H7, B2K8T1, B2SS66, B2TVF4, B2VBT8, B3PI88,
B4ESU4, B4SM82, B4SZJ8, B4TC49, B4TQU0, B5BC30, B5EUP9, B5F073, B5FP62, B5QX66, B5R762, B5XZ74, B5YRL5, B6ESC6, B6I7T0, B7LC58, B7LJY7,
B7M748, B7MQM9, B7NA75, B7NNK6, B7ULX3, B7V486, B7VH15, B8F713, B8GTH6, C0PWY3, P12998, P36570, P44422, Q02TR5, Q0A5W2, Q0T6I4, Q0TJS2,
Q0VMD1, Q1C946, Q1CFQ4, Q1I3N7, Q1QYD6, Q21FY4, Q2NUJ6, Q2P8F2, Q2SBD5, Q31E54, Q324B6, Q32I45, Q3BYN0, Q3K5P2, Q3Z408, Q47829, Q48CS2,
Q4K4T3, Q4QKR3, Q4UZN9, Q4ZMA9, Q57RG2, Q5DZH9, Q5H5R0, Q5PG49, Q609V1, Q66D66, Q6D3C0, Q6LPR3, Q7CH66, Q7MLU9, Q7N6Q6, Q7VL09, Q83S45,
Q87DT2, Q87QN5, Q88A97, Q88QX1, Q89AK6, Q8D8N0, Q8FJQ2, Q8PDF1, Q8X823, Q8Z892, Q8ZQQ7, Q9CJU0, Q9I617, Q9KSZ3, Q9PDM2, P06986, P10369,
Q4QN73, Q5H0L0, Q65S79, P0AB77, P37419, P43336, P04693, P74861, Q73KM3, O07587, O33822, P36692, P53001, Q55128, Q56232, Q59228, Q60013,
Q82DR2, P63499, P9WQ90, P9WQ91, B0K590, B0KC20, B1XL23, B7K0L9, P74770, Q119K2, Q3M9A4, Q67N86, Q8DJ97, A0PP02, A0QHJ9, A0RIB9, A1KJ00,
A1T8U6, A1UHM3, A3DBD5, A3Q146, A4T9L3, A5U2S6, A5WMQ5, A6TU88, A7GSE1, A7Z5B4, A8FDG9, A8MEX7, A9VG56, B0C205, B0JQZ0, B1I4F9, B2HQ90,
B2J1W1, B7GHW7, B7HAZ0, B7HNN4, B7IWN1, B7JLX2, B7KD70, B8HTV6, B8ZR84, B9IWY0, O31777, P0A4X5, P22806, P45487, P53556, P9WQ86, P9WQ87,
Q1B7F0, Q5SHZ8, Q635G4, Q65ML1, Q6HE48, Q731H9, Q740R7, Q7NNL4, Q818X0, Q81MB0, Q8KZM9, Q8YZT3, Q9K625, Q8R5U4, P16524, A2CC97, A5GIN1,
Q0ID68, Q7V4Z3, P17731, P73807, Q02135, Q9KJU4, Q0S962, Q8FU28, P60120, P60121, P9WQ88, Q795M6, A6LMP4, A7HMM1, A9BGL0, B7ID58, O66630,
A6L7E4, A6L8U2, B6YRL2, Q5LC03, Q64SY6, Q8AAB8, A0LEA5, A1ATI6, A1VDD3, A5GD93, A8ZXV5, B3E933, B5EGX2, B8DJJ6, B8IZX8, B9M384, C0QFJ4,
C6BUK3, C6E9Q7, Q1MR87, Q30ZX9, Q39Z65, Q3A1U5, Q72BI1, Q74GT3, B0B7W0, B0BC25, Q253K9, Q5L6M0, Q6MDE0, Q824A4, Q9PK04, B0SEH8, B0SMK7,
Q04UL5, Q04YV8, Q72NJ3, Q8F814, P63503, P9WQ82, P9WQ83, Q08432, Q5SHW0, Q93QC6, Q9CBM9, Q84CG1, Q3MAL4, Q3MDN5, Q8YM38, Q8YP73, A2BT75,
A2BYM6, A2C4T7, A3DK17, A3PEY9, A5FRC5, A5GW23, A9BCJ1, A9KJ19, B0CDH5, B0JUM0, B0TA38, B1I544, B1WSG7, B1XKF6, B2A250, B2J2U3, B7JVL5,
B7KL61, B8CX89, B8HJY4, C4Z4Y1, C4ZG66, Q10ZC3, Q24S01, Q2JLL9, Q2JS04, Q2RK33, Q318P3, Q31PY6, Q3AC10, Q3AMU5, Q3AW44, Q3Z8H5, Q3ZXC8,
Q46IX2, Q55828, Q5N492, Q7NDX4, Q7U4C3, Q7UZZ3, Q7VA14, Q8DH57, P9WPZ4, P9WPZ5, Q02636, P39643, A7Z4X1, P26505, Q04512, Q06965, Q1RIV2,
Q4UJV4, Q68VS3, Q9ZCB8, O31665, Q8DTM1, B4U9L1, O67857, A0M287, A1BGB4, A4SE60, A5FFY0, A6GY79, A6L2V8, A6LAM2, B3ECG2, B3QP11, B4S8L6,
B4SGL8, Q11VM5, Q3ARM7, Q3B3L3, Q5LAZ9, Q64RE8, Q8ABA8, A3PIA4, A4WUN9, A5FVN2, A7ICA9, A8HZS2, A8LK96, B0UN04, B2IDA4, B3Q8Z5, B6IYQ0,
B8IRU5, B9KPH4, P45358, P55683, P61002, Q07IG8, Q0AM22, Q0C348, Q11DR9, Q131B9, Q163G3, Q1GET3, Q1GP30, Q1QQD5, Q20YH9, Q28TL1, Q2GAI1,
Q2IS68, Q2N7G6, Q2RP86, Q2W047, Q3J445, Q3SV41, Q4FP52, Q5FQA6, Q5FRR4, Q89GX0, Q89UL9, Q8U9W3, Q92L21, Q92MG0, Q930J0, Q987C8, Q98B00,
Q98G10, Q9A5B6, A1KV06, A1TKZ0, A1VK38, A1W431, A2SE05, A9M185, B4RJ05, B9MDV4, O07131, Q2Y6Y6, Q2YAU6, Q39CT7, Q39K90, Q39M27, Q3JMZ7,
Q3JW89, Q3SI68, Q3SK85, Q46Y48, Q47AL9, Q47GP2, Q5F7D7, Q5P791, Q62FC0, Q62GE0, Q63Q87, Q63XM1, Q7P0F4, Q7VSZ0, Q7VWL5, Q7W2Y3, Q7W6Q1,
Q7WDY3, Q7WHN5, Q82WM3, Q82XE0, Q845V2, Q8Y0Y8, Q9JTH8, Q9JYH7, A0RMN9, A1VEW4, A1VY36, A5G9G1, A6Q1Z5, A6QBY8, A7H084, A7H556, A7HCR6,
A7I2V8, A7ZCF3, A8FKA6, B5E9W9, B9KDN6, B9LZ53, C6E916, P61000, Q2LST8, Q30TC9, Q311Z4, Q39YP6, Q3A7R3, Q5HWF4, Q6AQK2, Q72DA0, Q7M7Y6,
Q7VIJ3, A0KKB7, A1ACN3, A1JTV9, A1S6Z2, A3M2I8, A4SMP7, A4TKK4, A4WC70, A5CVR5, A5F2A2, A5VZ57, A6TBC4, A6VUD3, A7FJH1, A7MJP4, A7MX17,
A7ZNJ3, A8A1P5, A8AEK3, A8GC78, A9ML15, A9MSC2, A9R2K5, B0KQJ6, B0RSL5, B0TY45, B0U3B2, B0V7Q2, B0VV21, B1IZ53, B1JBC0, B1JPW1, B1LP20,
B1X6V8, B2FPM0, B2HTW5, B2I5Y0, B2JZM8, B2TYF9, B3PCJ2, B4STN8, B4SX42, B4T9N5, B4TMR6, B5BFB9, B5EX40, B5FDA0, B5FM42, B5QZL3, B5RBR3,
B5XPE6, B5YU77, B6EJ89, B6I848, B7GZI3, B7I6C5, B7L9P8, B7LUF2, B7M400, B7MDH5, B7MWU0, B7NC61, B7NQG9, B7UT58, B8D707, B8D8Q3, C0Q1K1,
C3LU31, C4ZSB0, C6DF75, P44423, P57202, P58891, P58892, Q058A6, Q0T3A6, Q0TG66, Q15RU8, Q1C9R1, Q1CGX0, Q1IE97, Q1LT68, Q1R089, Q1RA52,
Q2NTX2, Q2P3K2, Q2SBJ7, Q31GD4, Q31I36, Q323J1, Q32EF0, Q3BUF6, Q3J7H2, Q3JEN8, Q3K8U2, Q3KHZ1, Q3Z0G4, Q47XB7, Q48ED0, Q492K2, Q4FQF9,
Q4FSH2, Q4K8N0, Q4KI72, Q4QLD1, Q4UU41, Q4ZNW0, Q57004, Q57MS2, Q5E637, Q5PDP4, Q5QWQ9, Q5QZ49, Q5WX92, Q5X5X0, Q5ZW88, Q608S3, Q609W4,
Q65RB2, Q66C50, Q6D410, Q6FEC7, Q6LT75, Q7MLS5, Q7VQW9, Q83KJ6, Q84I51, Q84I52, Q84I53, Q87C30, Q87QL0, Q87WV6, Q88P86, Q89AX7, Q8D8Q1,
Q8EFB2, Q8FG51, Q8Z5J9, Q8ZFX6, Q9CLM3, Q9CMI7, Q9KSX2, Q9L6I2, Q9PBC6, Q9RI00, Q9S5G6, Q9ZHE5, P60998, B2UPR9, B3DXN2, Q7UNC3, Q04QW8,
Q04Z75, Q72PG3, Q8F6W9, C0R1Z0, O52815, A0AK37, A0PP15, A0PXP5, A0QHI1, A0QX82, A1A2H6, A1R558, A1T8W2, A1UHK7, A2RKS5, A3CNT7, A3Q130,
A4IQ80, A4QFG6, A4XMY1, A5CZ78, A5FR29, A5I245, A5N7Q7, A5V022, A6LUF3, A7FU81, A7GDQ6, A7GN55, A7NFV2, A7Z614, A8AY31, A8FEJ6, A8MEH2,
B0JJJ7, B0K625, B0K735, B1HTD4, B1ILA9, B1N009, B1WY56, B2GBR8, B2HQA3, B7GHJ8, B7JUI4, B8DC01, B8FP20, B8HW95, B8I5V1, B8ZRB0, B9DK21,
B9E168, B9EAC1, C0ZCE7, C1ATZ5, C1FN41, C1KWM5, C3KVX5, C5D3D2, P16246, P28735, P60999, P61001, Q02YW3, Q03K75, Q03VY3, Q0SHX9, Q10VS0,
Q1AY33, Q1B7G5, Q24QJ1, Q2J8K9, Q2JPM4, Q2JTG5, Q2RL44, Q2YSI3, Q31PF9, Q3AAT6, Q3AD52, Q3M504, Q3MAX6, Q3Z879, Q3ZXL8, Q47QS8, Q49VS0,
Q4JW58, Q4L4E7, Q5KXV3, Q5N4R3, Q5WGR9, Q5YYP9, Q63A05, Q63DL4, Q65I37, Q67KI2, Q6A8L4, Q6GBA6, Q6GIR8, Q6HHF6, Q71Y90, Q736A5, Q73AX7,
Q7NL03, Q81C43, Q82AA5, Q88UE6, Q8CTG8, Q8DM42, Q8DTQ4, Q8EQB9, Q8ESS3, Q8FNZ1, Q8G4S8, Q8KZ92, Q8NXN3, Q8R5Q4, Q8Y5X8, Q8YMG7, Q8YV89,
Q97ES6, Q9KCA8, Q9RRM7, Q9X7B8, Q72LL6, A5INE2, B1L869, B9K9R9, Q3S8P9, A0PVN0, A0Q9F3, A0R5X8, A1KQA5, A1TGS6, A1UN51, A3Q7J9, A4QAL4,
A5U9A1, B1MFC0, B1VP97, B2HLJ8, C0ZM44, C1AIM6, C1B997, P61004, P61005, P9WML4, P9WML5, Q1B1Z8, Q47KH1, Q4JSJ5, Q5Z3C0, Q6ABU3, Q6ABX6,
Q7TVQ0, Q82FJ1, Q8NTT4, Q9ZBY8, P95468, O85746, P77806, Q5HIC5, Q6GBT7, Q6GJB8, Q8EM07, Q8NXY3, P63501, P9WQ89, A5IQS7, A6QF32, A6TZK2,
A7WZL0, A8YZZ5, P67724, P67725, Q2FIR7, Q2G087, Q5HHU9, Q81FQ1, Q3J9D6, C6C2Z3, Q5HR08, B2JKH6, Q46WL3, Q9PII2, P58661, P0A959, P0A960,
P0A961, Q9KM65, Q9HVX0, P0AB78, P0AB79, P23034, P96847, A0QX65, A1KJ16, A5U2V6, C1ANM2, P0A679, P9WML6, P9WML7, Q92A83, Q9X0D0, Q9X0Y2,
A8EWM9, P00509, Q8Z8H8, A1A918, B7MGN4, Q1REF4, Q5WV43, Q81P62, Q8KD01, Q7VWP1, Q7W9I4, Q8PQD8, O84395, Q3KLW3, Q9A671, B1YMC6, Q81I05,
Q81V80, Q8XV80, Q5X3Q5, Q5ZU10, Q9HZ68, A5VSV7, Q2YR81, Q57AR7, Q8FY98, Q8YJK3, Q6HL37, Q81SV5, A5UA19, A5UGY2, Q9Z856, Q5KY23, Q18T09
```

Radical SAM (PF04055), referred to as SAM in Chapter 4:

B2V8G9, B4U973, C0QR28, O67104, B1GYW9, Q3ARP5, A0M7A9, A1BCQ5, A4SGW2, A5FLT1, A6GW77, A6LD84, B2RH08, B3EI42, B3EPW2, B3QLR7, B4S690,
B4SGZ7, Q11S94, Q2S4I8, Q3ANX4, Q3B169, Q7MT97, Q8KGB6, Q02550, B5ELR8, B7J403, B5YK85, O31381, P33770, P95651, A3PH74, A5VAC6, A6X2S8,
A7HP26, A7IEC3, B3QCX3, B4RBP5, B6JDD7, B9KM96, Q07PI4, Q138Z3, Q1GTT5, Q212A8, Q2GAF7, Q2IUT3, Q3J561, Q3SW30, Q4UM45, Q5LN74, Q5NRD6,
Q6N859, Q8REU0, A0L3M0, A1B1Z0, A4YQS3, A5EFG5, A5FZN9, A8IJU8, A9CFX5, A9HRF2, A9W8M8, B0T1Y4, B0U811, B1LV19, B1ZFX7, B2IEZ6, B7KN34,
B8EMZ5, B8H640, B8IU36, B9JYY6, B9KGM2, Q0ATN3, Q0BUV5, Q0C661, Q1QRH1, Q2GDF4, Q2GHB1, Q2GLB4, Q2NB65, Q2W3L4, Q3YRG6, Q5FFY2, Q5FPC9,
Q5HAN0, Q5PA14, Q9A2N5, Q9AMS7, P07748, A8GUH3, Q0BSW6, Q1GGW2, Q2K8V9, Q4UK97, Q92GH8, Q9RNY7, A7IGB2, A9HIM2, B6JCT6, B7KVE0, Q5FHD1,
Q5GSS2, Q5HBR5, Q5LLM0, P06770, P09824, P24427, P71517, Q89FG1, Q9L3B0, A4WRD4, Q6NCS3, A8LNF0, B1M1U6, Q28VS6, Q2K3B1, Q2W897, Q3SNT2,
Q7CWI1, O87941, A0KB05, A1K9C8, A1KU01, A1TQ53, A1VUJ4, A1W7J6, A2SGQ6, A4G1F1, A4JIB7, A4SV63, A6SU66, A9AE44, A9C2R2, A9I023, A9LZ69,
B1JZE1, B1Y502, B1YNS2, B2AGA0, B2JKH4, B2SWS5, B2UDA1, B4E9L4, B9MJH4, P94966, Q0AE72, Q0BBD4, Q0KF86, Q12D73, Q12F39, Q146K5, Q1BT34,
Q1GZA6, Q1LS73, Q21W43, Q2KWF1, Q2Y9Y9, Q39CE4, Q3JWR8, Q3SLY0, Q477A1, Q47IF6, Q5P7M6, Q7NPW1, Q82SL7, Q8Y2R9, Q9EYP3, A9HZZ2, Q39D51,
Q3SFF1, O34162, A9AH21, A9BMV6, Q82XV4, A1K1N5, Q5P119, Q30Y73, A1VF04, A6QCD0, B2USA0, Q17YC7, Q7VGP6, A0LJA8, A1AN77, A1VB97, A1W1T3,
A5G3Q7, A6Q5K0, A6Q6D8, A7H0I4, A7H632, A7HG97, A7I1A0, A7ZFR4, A8EWX7, A8FNZ2, A8ZUS9, A9EXH2, B2UVE8, B3E599, B4UCB2, B5ECN1, B6JNP9,
B8DPP9, B8FHQ1, B8J0C6, B8J638, B9KEN8, B9L7E1, B9M305, C6C0T2, O25956, Q0P7U6, Q17VA0, Q1DCV9, Q1MRA1, Q2IF61, Q2LY09, Q30U81, Q30XZ5,
Q39VB0, Q3A4K3, Q5HSB2, Q6AK48, Q728P5, Q74CT7, Q7M8P7, Q7VFE6, Q9ZJK8, A7GZE1, A7ZE07, A7H5G3, Q2IM41, Q1D4N0, Q2IIW6, Q57390, Q2IHD1,
Q9X758, A0KIC6, A0KY79, A0Q638, A1AWE7, A1JS69, A1RIK5, A1S5I9, A1SW31, A1U4B2, A1WVM7, A3D3F2, A3MYI9, A3QDN8, A4IXP5, A4SPR7, A4TNQ6,
A4VR88, A4W8B7, A4XZR9, A4Y7Y3, A5CWW0, A5F2H3, A5IBW2, A5UD65, A5UIF3, A5VXF1, A5WHI6, A6UYW0, A6VNF1, A6W0Y1, A6WM55, A7FKM9, A7MJ03,
A7MX36, A7NCW7, A7ZJI4, A7ZY31, A8AJ12, A8FX10, A8GBC5, A8H3I7, A9KG71, A9KY60, A9MJE6, A9MTI8, A9NCS0, A9R3C8, B0BS22, B0KJ53, B0RMR2,
B0TJN8, B0U0S3, B0U2A0, B0VR41, B1IXJ3, B1JE55, B1JSS4, B1KPJ7, B1LM66, B1WN69, B1X7A5, B2FLM4, B2I672, B2K8T0, B2SGE3, B2SS67, B2TVF5,
B2VBT9, B3GZV8, B3PI87, B4ESU3, B4S0P9, B4SM81, B4SZJ7, B4TC48, B4TQT9, B5BC31, B5EUQ0, B5F072, B5FP61, B5QX65, B5R761, B5XZ75, B5YRL4,
B6ESC7, B6I7S9, B6J074, B6J725, B7LC57, B7LJY8, B7M747, B7NA74, B7NNK5, B7ULX2, B7V485, B7VH16, B8CQY2, B8D7I5, B8D983, B8EAJ2, B8F6C9,
B8GTH4, C0PWY2, P12678, P12996, P36569, P44987, P57378, Q02TR6, Q084I8, Q0A5W1, Q0BLD5, Q0HHN7, Q0HTY9, Q0I355, Q0TJS3, Q0VMD0, Q12NN4,
Q14HR4, Q15SR5, Q1C947, Q1CFQ3, Q1I3N6, Q1LTL8, Q1Q8S6, Q1QYD5, Q21FY3, Q2A2V9, Q2NUJ7, Q2P8F3, Q2SBD4, Q31E55, Q324B7, Q32I44, Q3BYM9,
Q3IGS6, Q3J9D5, Q3K5P1, Q3Z409, Q47862, Q481G0, Q4FQJ9, Q4QLQ0, Q4UZN8, Q4ZMA8, Q57RG3, Q5H5R1, Q5NGB2, Q5PG48, Q5QZ16, Q5WW97, Q5X592,
Q5ZVG8, Q609V2, Q65SD0, Q66D67, Q6D3B9, Q6FAP9, Q6LPR2, Q7CH65, Q7MLV0, Q7N6Q7, Q7VMH0, Q83CU5, Q87F85, Q87QN6, Q88QX2, Q89AK5, Q8D2A1,
Q8D8M9, Q8EDK6, Q8K9P1, Q8PDF0, Q8PQD7, Q8X825, Q8Z893, Q9CNP8, Q9I618, Q9KSZ4, Q9PH80, Q44634, P32131, P77915, P60716, P60718, Q0I1H8,
Q0T6N8, Q31F42, Q5QYE5, P10390, Q8ZDH0, A6T6B5, A8AJE9, B0BQR0, B2VBL4, P0AEI3, Q0ABN9, Q1QV24, A1S8S1, A4VQY3, A4XYW1, A9MKB7, A9R6X8,
B1JG95, B2K896, B2SW86, B5BCD1, Q0WDR2, Q1C531, Q1CKP3, Q3IK75, Q51470, Q5GZ51, Q5PLT0, Q87AP4, Q8PJR9, Q7MM75, Q9HXD6, O33506, P07782,
Q6F9I9, Q5H234, P64555, A1S2T9, A4VKC4, A8AIT5, B0RR62, B2SMB3, Q3BTW5, A1RHQ0, A4VNX4, Q0I3U8, Q15R53, Q51385, Q8XAA4, B2UNW0, B3DV36,
Q255G8, Q5L5F9, Q7UF84, Q9Z6L5, Q9Z8T3, B1ZVI7, B0SDC5, B0SLQ3, C0QVM0, Q04RS9, Q051U2, Q72RG8, Q8F498, P71011, B0JTF9, B2IW30, B8HP19,
P72811, P73191, Q113H3, Q11A09, Q31PU7, Q3MCR5, Q3MF37, Q5N4D1, Q7NIT2, Q7NMI9, Q8DII8, Q8DIR6, Q8YPR3, Q8YR77, A0JTA3, A0LTP8, A0PP05,
A0PYU9, A0QHJ1, A0QX70, A0RIB6, A1KJ05, A1R3R7, A1SM80, A1T8V0, A1UHL9, A2BRS2, A2BX80, A2C3I4, A2C8D5, A3DBD3, A3PDJ8, A3Q142, A4FIS6,
A4IN84, A4J1M3, A4QA10, A4T9L9, A4X585, A4XGB8, A5D4Y6, A5GLZ1, A5GS37, A5I427, A5IVK5, A5U2U5, A5WMR0, A6LUQ4, A6QJR4, A6TT61, A6U4F5,
A7FVS6, A7GFJ9, A7GSD8, A7X669, A7Z5B2, A8FGR8, A8G5G3, A8L2N0, A8LUR1, A8YYH5, A9BB12, A9VG53, A9WL37, B0CC69, B0JFP4, B0TE53, B1HRT3,
B1I4G4, B1IHH9, B1KW08, B1MBZ3, B1VF77, B1W5F4, B1WTI3, B1XNE1, B2GLQ7, B2HQ93, B2J914, B2TL73, B2V167, B7GKT8, B7HAY7, B7HNN1, B7IWM8,
B7K1U3, B7KFJ9, B8HDH4, B8HPP0, B8ZR86, B9DKN5, B9E2A2, B9EA22, B9IWX7, B9MP50, P0A507, P19206, P46396, P46715, P53557, P73538, P9WPQ6,
P9WPQ7, Q0IBC8, Q0RD46, Q0SHW6, Q0TQ59, Q10YQ3, Q18D35, Q1B7F4, Q2FE72, Q2FVJ7, Q2J6H8, Q2YVY6, Q31AD2, Q31R68, Q3ADP5, Q3AJ51, Q3AX82,
Q3K2Q5, Q3M4U9, Q46K32, Q4JWG3, Q4L9U7, Q5HDC9, Q5KZN1, Q5N332, Q5SKN6, Q5YYR3, Q65MK9, Q6G6P6, Q6GE08, Q6NHL3, Q6NKC7, Q70JZ1, Q72L21,
Q731I2, Q740Q9, Q7A018, Q7A3R9, Q7NDA8, Q7U7P8, Q7V101, Q7V6T8, Q7VBJ0, Q818X3, Q826T2, Q899M1, Q8CQB3, Q8DL38, Q8E197, Q8FUD1, Q8KZM7,
Q8XK59, Q8YVQ3, Q97MI6, Q99RK7, Q9FCC3, Q9KC26, C9XIS7, C9YHW3, A0RKF7, A4FA70, P61194, Q83MU9, Q8DL83, A9KMU9, B7GQG0, Q49573, P62589,
P9WJS2, Q44118, Q8NR60, Q93KD1, Q9WX96, Q0IC70, Q2JK17, Q2JSD0, Q31D78, Q5N1N6, A4X4J7, A5U6N5, P9WH15, Q8YZV0, O31423, A5IK57, Q1LNN2,
A0A384LP51, A7ZJ54, A7ZXT0, B1IYE7, B1LLB3, B1X656, B5YQL3, B7L9K8, B7LKU3, B7M5I8, B7N9R6, B7NM12, P0AEI1, P0AEI2, Q0TK19, Q32IR1,
Q2JRI4, Q8E6Q1, Q8KCU0, Q30W71, P20714, A0Q2E1, Q8XMQ3, Q55373, P39280, P44641, Q8ZKB8, C3K2L8, A5I5U3, A7FX96, A7GH77, A7Z1T2, A9KK15,
B1IL14, B1KZ37, Q5WJ42, Q185C5, A0A1C7D1B7, Q9X0Z6, B2V930, B4U8S7, O66638, O67368, Q1II13, Q01RU5, Q1IQH5, B5Y8R7, B5YE40, B1H0D2,
A0M4W4, A0M7D3, A1BFY3, A1BIX2, A4SCW6, A4SEG3, A5FIU2, A5FNF9, A6H119, A6KZJ2, A6LE09, A6LEM6, A8Z609, A8Z642, B2RKG6, B2RKU2, B3ECK5,
B3EGT4, B3ENP4, B3ES11, B3QNE9, B3QR49, B3QVA0, B4S564, B4SCB6, B4SHB4, B6YRD1, Q11Q84, Q11U26, Q3AU39, Q3B3Y5, Q3B6A6, Q5LGU6, Q5LJ70,
Q64XQ0, Q650P7, Q7MAW4, Q7MWT4, Q8A029, Q8A2W0, Q8KB05, Q8KDH2, Q8KC85, B5EMR4, B5ENG4, B7J5B2, B7J7N7, B5YKW2, Q8RG43, Q8RHX4, A5G2D2,
Q8UFG1, A0L509, A1B1T3, A1USA0, A3PJL9, A4WRC8, A4YUR5, A5CCM4, A5EIM5, A5V2U6, A5VQP8, A6U8F5, A6X0M7, A7HXW0, A7IM78, A8EZU2, A8F2M4,
A8GPR5, A8GTI8, A8I4L8, A9IS79, A9M5D7, A9W6I4, B0BV26, B0CGS4, B1LWN4, B1ZEL4, B2IB58, B2S5X5, B3CP16, B3CQP5, B3PYS0, B3QIL2, B4RBV1,
B5ZNB1, B6IQ36, B6JDY1, B7KRC9, B8EJU1, B8IDD6, B9JEZ6, B9JW84, B9KSE2, C0R2Q4, C0RJ95, C3PLL8, C4K1I0, O05959, P61198, P61200, P65281,
P65282, Q07MZ5, Q0APQ9, Q11HV6, Q137C6, Q1GTM7, Q1MH26, Q1QMA2, Q1RGX4, Q214R4, Q2G712, Q2GE84, Q2GGX7, Q2GKX7, Q2IW20, Q2NAD7, Q2RT68,
Q2YPV8, Q3J2P4, Q3SSP1, Q3YRT2, Q4FM35, Q57D15, Q5FFK4, Q5FNM1, Q5GSY2, Q5HB02, Q68W08, Q6G166, Q6G401, Q89LR6, Q89NW6, Q92Q94, Q98MY2,
A0L7K3, A1B8C4, A1UU39, A3PPW5, A4WZB3, A4YJD7, A5CC78, A5E855, A5FY82, A5V8Y0, A5VTA1, A6U5H0, A6WWX6, A7HZ82, A8F011, A8F2U6, A8GQ03,
A8GTT8, A8GY24, A8IG00, A8LSE7, A9IMW7, A9M9Y3, A9VYZ9, B0BVC7, B0CK00, B0T155, B0UNY0, B1LWE8, B1ZJR5, B2S9E5, B3CLG9, B3CV38, B3PZB6,
B3QAC6, B4RC70, B5ZMY1, B6IVM2, Q07V68, Q0AK79, Q0BVY6, Q11BD9, Q13EK7, Q161G5, Q1GK98, Q1GRE9, Q1MMB6, Q1QS74, Q1RKC2, Q21C43, Q28UJ8,
Q2G9P6, Q2GCU4, Q2GG00, Q2J2K9, Q2KD88, Q2N950, Q2RMT1, Q2VYQ8, Q2YQS8, Q3IW81, Q3SMT7, Q3YSK6, Q4UK06, Q57AB1, Q5FPF1, Q5NRF3, Q5PB69,
Q68VU1, Q6G0N1, Q6G4V6, Q6NCM4, Q73HW8, Q89W97, Q8FXU4, Q8YEA2, Q92G77, Q92SI7, Q98BK3, Q9ZCE8, P17434, Q53205, A1K1U7, A1TJ31, A1V016,
A1VIT9, A1W2S7, A1WF48, A2S5Y9, A3MR17, A3N576, A3NQX6, A4G9C6, A6T345, A9BPT7, A9LZ95, B1XX30, B2AG37, B2UE01, B9MB93, C1D5T3, C5CP21,
Q0AI05, Q0KFE6, Q1GYC1, Q1LSC9, Q2L1D8, Q2T1K5, Q2Y7I9, Q3JWL1, Q3SM23, Q477G5, Q47JD5, Q5P4B8, Q62N16, Q7NTF9, Q7W0K8, Q7W222, Q7WR00,
Q82UJ5, A0KAB6, A1IQ34, A1K3J0, A1KS33, A1TKI5, A1V713, A1VSP2, A1WBW2, A1WIL9, A2S8Q1, A2SD55, A3MNX8, A3N5Z1, A3NRN5, A4G267, A4JHN3,
A4T007, A6SV24, A9AFF8, A9BUQ1, A9M1G4, B1JY57, B1XS06, B1XYX5, B1YWB9, B2AHA4, B2JD88, B2SYI5, B2UFP3, B4EBG2, B4RNW8, Q0AF59, Q0BC21,
Q0KE93, Q12EH0, Q13UD5, Q1BTS3, Q1H3L6, Q1LR87, Q21T29, Q2KWE0, Q2T101, Q2Y5J2, Q3JVV1, Q475N7, Q47A75, Q5FAI1, Q5P760, Q62MK8, Q63X63,
Q7NQI8, Q7VZ86, Q7WB66, Q7WMN3, Q82SI7, Q8Y206, Q9JXV8, A0LFB7, A1APR6, A5G670, A6Q531, A9FST8, B3E424, B5EE49, B5ZA35, B6JKJ8, P56131,
Q1CUN4, Q1D5V7, Q1MQ52, Q2LT94, Q30PR1, Q39TA3, Q3A594, Q6ALW9, Q74B44, Q7MR25, Q9ZMG6, A0RQN5, A1VYH4, A7H4S5, A7HAH8, A7I0Q3, A8ERE9,
A8FKP3, A8ZVH2, B4UK35, Q0PB55, Q2IKE9, Q30YS1, Q5HW09, A1AMT5, A1VF79, A5GCS6, A9FD62, B3E6X0, B5EAX0, B9M4F4, C6DYX9, Q30XT6, Q39QW1,
Q6ARK0, Q72DM4, A0KNA3, A0KTV4, A0Q6Q2, A1A8Q3, A1AWI8, A1RGT2, A1S8U1, A1SZ17, A1U380, A1WT98, A3D7P9, A3N2P1, A3QH60, A4IY90, A4SJV5,
A4VQZ6, A4W814, A4XYX3, A4Y9G2, A5CWR6, A5EVQ3, A5F2Y0, A5IGG3, A5UBB2, A5UFJ8, A5W9I5, A5WDV8, A6V0B4, A6VM79, A6WRL2, A7MNP9, A7MY96,
A7NBV0, A7ZJ15, A7ZXQ4, A8AJH6, A8FZ23, A8GB09, A8H7D5, A9KFW5, A9L014, A9MKE4, A9MV74, A9NDX8, B0BRR9, B0RNP9, B0TR57, B0TZD6, B0UVP7,
B1IYI0, B1J144, B1KDX3, B1LKL9, B2I923, B2SH72, B2TTI9, B2VBK1, B3H2J0, B4EV01, B4SJ16, B4SYJ0, B4TAI5, B4TPA3, B5BCG1, B5EZ72, B5FBJ6,
B5FMM8, B5QVN4, B5R7Y1, B5YQH9, B6EIF3, B6I137, B6IZL4, B6J7S3, B7L9H2, B7LLI3, B7M5F5, B7MFQ1, B7MRR5, B7N9N3, B7NLY7, B7UKR9, B7V9C1,
B7VKE6, B8CSH7, B8D7G6, B8D962, B8E4W4, B8F679, B8GMV6, C0PW62, C1DMR1, C3LTJ1, C4LCC5, C4ZWB5, C5BGD9, C6DBV6, P44463, P57357, P57978,
P60717, P60719, Q02SG3, Q057Q7, Q087L5, Q0A8A4, Q0BM68, Q0HLK3, Q0HXV7, Q0TK45, Q0VN36, Q12QX1, Q14IH3, Q15VL3, Q1I4G1, Q1LTM3, Q1QCS6,
Q1RET3, Q2A3R2, Q2NUV8, Q2NYZ2, Q2SA38, Q324R5, Q32IV2, Q3BXQ3, Q3IJ81, Q3J7W4, Q3K6A4, Q3Z4G4, Q484R8, Q4FTQ5, Q4QPL8, Q4UYT2, Q57RU3,
Q5E6V9, Q5GVR7, Q5NH21, Q5PM97, Q5WYF5, Q5X703, Q5ZXI6, Q60CJ6, Q65RH7, Q6D7M8, Q6LN87, Q7MN17, Q7N767, Q7VKB1, Q83C63, Q87RR1, Q87VW7,
Q88DM5, Q89AL7, Q8D325, Q8DFD1, Q8EHQ6, Q8K9Q2, Q8P590, Q8PPL8, Q8Z8I3, Q8ZR04, Q9HX25, Q9KTF9, Q9PDW0, A1JPR9, A4TP09, A7FKY9, A9R703,
B0U641, B1JGB7, B2K874, Q1C507, Q1CKR6, Q66DF5, Q87DZ9, A0KN81, A0Q6T5, A1A8S9, A1U494, A1WVF7, A3N1X3, A4IYC1, A4SJX4, A4W838, A5EXA7,
A5IBH7, A5UBB9, A5UFJ1, A6VQQ3, A6VZE1, A7MQS5, A7NBR0, A9KCP2, A9NC58, B0TXS4, B0UUU9, B0V6R4, B0VQX2, B2HZ55, B2SG13, B3GYA4, B3PER6,
B4S0R8, B5FBL3, B5XZQ0, B6EIN8, B6HYN0, B6J1A4, B6J853, B7GW55, B7I9D5, B7MFS7, B7MPH7, P57516, Q057G5, Q0BM96, Q0I3Z1, Q0VN66, Q14IK5,
Q1LU21, Q1REQ5, Q2A3U6, Q2SBF4, Q31IF4, Q3JEH9, Q47Y80, Q4QPM5, Q57163, Q5E6U2, Q5NH53, Q5QYC5, Q5WX12, Q5X5N2, Q5ZVV6, Q608N1, Q65RW3,
Q6D7K6, Q6F7W8, Q7MB63, Q7VP74, Q83DX3, Q89AC2, Q8K9C2, Q9L699, A0KTX4, A1AX14, A1JQA3, A1RGV4, A1SSC4, A3D7M8, A3QH41, A4Y9E1, A5CW77,
A5F2X6, A5WGM4, A6V0C9, A6WRJ1, A7FKU9, A7N1G9, A8FZ02, A8GB34, A8H7B6, A9KZZ3, B0RRW2, B0TR38, B0U4P1, B1KDV4, B2FJ92, B2I8U1, B2TU59,
B4EV28, B4SQK7, Q087J4, Q0HLI3, Q0HXT7, Q0T6S0, Q12R22, Q15TR4, Q1Q929, Q21FH4, Q2NUT9, Q2P266, Q324N4, Q3BS94, Q3K6B7, Q3Z4D1, Q48DN8,
Q4FQU4, Q4UVS2, Q4ZN97, Q66DD3, Q6LNA5, Q7MN00, Q83LY3, Q87RP4, Q87VY1, Q8CX45, Q8DFE8, Q8P8B5, Q9KTE0, Q9PEX2, Q6MLC6, B2UQE7, B3E0M0,
Q6MAB7, Q7ULM9, B0B8D2, B0BA11, O84562, Q256D6, Q3KLD9, Q5L4V4, Q6MEX3, Q7UH37, Q821K6, Q9PJI2, Q9Z774, B0S9E2, B0SS31, B2S3Z3, O83735,
P61199, Q04T63, Q04UA3, Q051I5, Q053R5, Q72PA4, Q72RU2, Q73MD3, Q8F3V7, Q8F743, E3PRJ8, Q9XBQ8, A0LTD9, A0PNH7, A0QEY6, A0R075, A1SJ19,
A1UIB4, A3Q1S8, A4ISG5, A4J247, A4QFS3, A4TBK7, A4XHV2, A5CQ99, A5GW06, A5US48, A6TWC9, A6WD63, A7GUG3, A7Z8E8, A8FH34, A8G714, A9VNX6,
A9WEM1, B0RE24, B1VHG8, B1VZM3, B2GJ88, B2HHL4, B7HCZ6, B7HUW3, B7IMZ7, B8G782, B8ZQJ9, B9DIX8, B9EAJ9, B9J3H2, B9LM70, B9MQ24, C1EY56,
C3PHK6, C4LI04, C5C4Y6, C5CCN0, C5D7H5, O32129, O32962, P61197, P72980, P73572, Q0RFE8, Q1AT13, Q1B6R1, Q1IXJ7, Q2J897, Q2RHM6, Q2YWR5,
Q47R88, Q49W64, Q4JWE0, Q5HQN7, Q5SLQ4, Q5WDS6, Q5YZ59, Q631Z7, Q67MF5, Q6A9X0, Q6AFG7, Q72GV1, Q72YB2, Q7NFJ9, Q7NLU2, Q7U4D9, Q7U7Q2,
Q7UZY1, Q7V0Z8, Q7V507, Q7V6S0, Q7V9Z9, Q7VBJ3, Q816A0, Q82AP7, Q83NN8, Q8CPW4, Q8ERL8, Q8FNP4, Q8NNJ0, Q8R9E1, Q8YWC1, Q8YXD1, Q9RWA4,
Q9S2P2, A0LV00, A0Q0M5, A1SNG0, A2BSM6, A2BY12, A2C471, A2CB74, A3DDI9, A3PED9, A4FAJ1, A4J5Q8, A4X4V8, A4XL48, A5D2K1, A5FPV2, A5I2S3,
A5UUG7, A6LWI1, A6TR80, A7FUL1, A7GE46, A8G6B6, A8L6J8, A8MFD5, A9AZY8, A9BBS2, A9WHB1, B0C0E2, B0JVM6, B0S1C0, B1I241, B1IM69, B1KSA4,
B1WU96, B2A3X6, B2IT24, B2TIA8, B2V276, B3DQX6, B7JZ48, B7K993, Q0RDX2, Q0STS9, Q0TRE5, Q1IYM0, Q24X58, Q2RJG3, Q319I4, Q3ACA4, Q3Z6Q4,
Q3ZYS0, Q46JG6, Q6A8Z7, Q7NE65, Q7V0F7, Q7V8L8, Q7VAS5, Q895H1, Q8EUX4, Q8G4H4, Q8RA72, Q8XLE9, Q8YXA3, Q97I18, Q0TPG6, P9WJ78, P9WJ79,
A5GML6, A5GRJ8, B0K1A1, B0K9L4, B1XQK7, P73127, Q0AYB7, Q119H9, Q2JKY0, Q2JQX6, Q31KL5, Q3ALB4, Q3AV90, Q3M8N9, Q47RR8, Q5N199, Q5SME9,
Q67NJ9, Q72J39, Q7U5R0, Q83FT9, Q83HG3, Q8DJB2, P20627, Q43883, C0ZY23, A5IJD4, A6LJ47, A7HK86, A8F716, A9BGV7, B1L8F3, B7IFC4, Q9WZC1,
O67886, P51008, Q51676, O25376, Q9ZLH0, Q38HX2, Q84F14, P74132, Q53U14, O66772, O67826, O67929, Q1IHK7, P59038, A6H1N2, Q89ZC3, Q8KFK8,
B5YJ09, A0L7R4, A1B4A2, A5VQC9, A6U9U3, A8LIR8, A9MAX5, B0CLT0, B2S5I1, B3PQ08, B3QCQ7, B5ZRM8, C0RIT4, Q139F2, Q1QN98, Q2K7L4, Q2NCE3,
Q2YNT7, Q3STB0, Q3V7S1, Q57DG3, Q89NT2, Q8G0X4, Q8UER0, Q8YGY6, Q92PB4, Q98MK6, Q9AC48, Q9X5W3, A9W3R0, B0UE13, B1LUE7, B1ZIB2, B7KXC6

SAM dataset continued:

```
B8ENI9, B8ITV7, Q0BQS8, Q5LTB4, Q6N8F3, Q9EXU8, Q9ZDB6, A9CF16, Q2RSY6, Q9ZCV2, Q8Y0K4, Q9K0Q5, A0RMJ2, A1VXP5, A6Q5D0, A6Q6R2, A7GWA5,
A7H1N9, A7ZB87, A8ERS0, A8FJX0, B6JM01, B9KDV3, B9L851, C6C060, P56414, Q1CTA2, Q30P92, Q5HX04, Q747W9, Q9PIW6, Q9ZL75, Q74CF3, A0KIL8,
A1SWQ3, A1U578, A3N054, A4XW05, A5F2Q5, A5UBK4, A5UFB4, A5VZZ2, A6T6M5, A6VPY2, A7MJ10, A9MTH7, B0BNX2, B0KST1, B1JCW0, B2FUM0, B2TVE9,
B3GXC5, B4SL67, B4TC55, B5XYW6, B7LJY1, B7ULX8, B7VLU4, C3LTS1, C4LFK3, C5B7H2, P45311, Q12T28, Q1I6J5, Q21I62, Q31JB9, Q3KAS8, Q4QJR9,
Q4ZU05, Q65TT2, Q6D3C7, Q7N6P6, Q7VLN1, Q87MY0, Q882V6, Q88E69, Q8D894, Q8PNH1, Q9CN21, Q9I3K7, Q9KT81, A4XTR4, A5VXG4, A6T9H2, B0V494,
B0VQD7, B2I0I5, B2VL10, B5XX58, B7H2Y0, B7I6M8, B7VAB6, C1DEW5, P27507, P59748, Q01060, Q02LD6, Q48CT3, Q4K4U8, Q4ZMC1, Q608P0, Q88A84,
Q88QV8, Q8PHY1, Q9I2C0, B0RQ25, B2SSW3, Q2P4Y9, Q3BQI4, Q4UXI4, Q8P6M7, P43751, P45097, Q8K9D9, Q7UVG8, Q7UT69, Q9Z874, O83293, P54462,
A0AHG0, A0PKZ7, A0R443, A1ULP7, A3Q648, A4J6S5, A4QDF8, A5CYZ0, A5I365, A5IV50, A6QJA8, A6TVF9, A6U3Z2, A7FUZ6, A7GEQ5, A7GU01, A7X5J1,
A7Z9P0, A8MLW5, A8Z366, B0CDZ6, B0JNC2, B0TI16, B1IN35, B1KU20, B1X0G3, B1XLR4, B2GCN4, B2HFA4, B2J1M7, B7JWW6, B8HWW4, C0Z9B3, C1B1N5,
C1FPG7, C1L1W8, C3KXJ8, C3LA56, P39757, P62588, P65385, P65388, P65389, P69848, P9WJS0, P9WJS1, Q0AVU6, Q0S2N2, Q119N9, Q1B3F3, Q2FEM4,
Q2JI46, Q2RGL2, Q2YYS8, Q3ADX8, Q3AGB7, Q3AVP9, Q3MC34, Q49ZI6, Q4L8D2, Q55369, Q56211, Q5HDT9, Q5HLY1, Q5N5F7, Q65DY5, Q6G754, Q6GEG6,
Q721B9, Q7NCF5, Q7TX84, Q7U3H2, Q88WY1, Q8CNE6, Q8Y870, Q8YQG6, Q92CY2, Q931G4, Q97HL8, Q9EYN8, Q9K9W9, Q9RJ47, Q9ZIM6, Q5SK48, Q8CJT5,
Q9K864, Q9XAP2, P59749, O68575, P0A442, P0A443, Q2FK43, Q2G1D7, Q2YV52, Q46267, Q5HJF3, Q6GCP9, Q6GK89, Q71ZR3, Q7A1W8, Q7A7X5, Q99WZ6,
Q81G67, P73667, Q7NCE3, Q7V9H9, Q97D55, A9BF51, Q9WZT7, B2V8N8, B4U6U1, O67016, Q028J0, O66761, B5YF65, B1GZH1, B2KB59, A0M3K8, A1BHA0,
A4SFH7, A5FA30, A6GZF6, A6L5E7, A6LAJ6, B2RHD7, B3EDL2, B3EPX5, B3QMN0, B3QSS3, B4S8Z6, B4SBD5, Q11XC6, Q3ASP1, Q3B317, Q5LCF8, Q64TK5,
Q7MXM3, Q8A9A2, Q8KCL7, B5EK28, B7J3M4, B5YKD1, Q7D1M2, P55477, Q136A2, A1B3K8, A3PIG0, A4YUI3, A5EJ58, A5V3Z4, A5VUQ0, A6X529, A7INV0,
A8I0G1, A8LI17, A9IUA4, A9MBK6, A9W8D2, A9WYN5, B0UQE6, B1M6H4, B1ZEV4, B2IHC3, B2SB89, B3QIV5, B6IRQ0, B6JF93, B7L2K9, Q07M57, Q0C0U1,
Q136X7, Q169Q9, Q1GIY4, Q1GV83, Q214L6, Q28TN0, Q2GA32, Q2IW68, Q2NDN5, Q2RQI7, Q2W2S4, Q2YKI5, Q3J3Y8, Q577W6, Q5LUV8, Q5NPC9, Q6G5E0,
Q6N6E2, Q89LG9, Q8FW94, Q8YC29, Q98MN9, A1TMP4, A0K7Q6, A1K487, A1V4H3, A1VQ18, A1WAJ7, A1WMV5, A2S2C9, A2SG87, A3MK46, A3NA26, A3NVU2,
A4G6D4, A4JEL4, A4SXC8, A6SXU1, A9BVZ2, A9IFP1, B1JT65, B1XUT6, B1Y223, B1YR17, B2JG80, B2T3U5, B2UFU5, B3R4X7, B4EAF5, Q0AG95, Q0BEZ5,
Q0KBP2, Q12A25, Q13Z56, Q1BGU5, Q1H0X3, Q21X03, Q2SWB9, Q2Y7J7, Q39FU0, Q3JRT1, Q3SI16, Q472G1, Q47CF0, Q5P1L3, Q62JZ1, Q63UQ8, Q7NYA1,
Q7VS95, Q7W1U6, Q7WQS2, Q82VY8, Q8XYX0, A2SLX7, A0RQM9, A1ATL9, A1V9Z2, A1W162, A5GBX9, A6Q526, A6QCC6, A7H6G8, A7I0P9, A7ZE21, A8ERB7,
A8FNC1, A9A0B5, A9F1Y8, B2UT98, B3EAM2, B4UKQ2, B5ED60, B5Z796, B6JLW7, O25434, Q0P8G1, Q17XY7, Q1CTD7, Q1DC90, Q1MQJ5, Q2LQ68, Q30PS0,
Q315T9, Q39QQ6, Q3A8J5, Q5HSX7, Q6AQ27, Q726F7, Q747R0, Q7VGL0, Q9ZEN7, Q9ZLA9, A7HIL1, A9EPV3, P56130, Q1CYE1, Q9ZMF0, A0KHZ9, A0L1C8,
A0Q4U9, A1A974, A1RNY7, A1U488, A3D958, A3N2T4, A4SKK7, A4XT11, A4Y2Z8, A5F514, A5IGM4, A5VZS9, A6VA58, A6VLD6, A6WIP6, A7MF19, A7NA32,
A7ZJQ2, A7ZY97, A9KZF7, A9MIP6, A9MSQ9, B0BRW2, B0KTL0, B0U054, B0US28, B0V6E8, B1IXD2, B1JD88, B1LMD0, B1X7X6, B2FP10, B2SWC0, B2TV93,
B3H2N3, B3PGP3, B4F137, B4RYE6, B4SQX0, B4TCV0, B5R840, B5XYQ3, B5YSC6, B6I8F5, B7H1U1, B7I9V4, B7LCB8, B7LMZ9, B7M7B0, B7MGU2, B7MQT8,
B7NAI1, B7NNR8, P0AEI4, P0AEI5, Q02I76, Q0A8I9, Q0BNJ1, Q0HEK0, Q0HZF3, Q0I4D9, Q0T6C8, Q0TJL3, Q15UT4, Q1I6D1, Q1RE90, Q21FE3, Q2P0S4,
Q2SBG3, Q31G14, Q323V7, Q3BRJ5, Q3KH22, Q3Z3U9, Q4KHA5, Q4UWF3, Q4ZWM9, Q5GXP5, Q5PGP7, Q5WYL5, Q5X765, Q5ZXP6, Q60CM4, Q65QA6, Q6D3N6,
Q6FCH4, Q7MG88, Q7VKK2, Q83LT1, Q88NL0, Q8D4N8, Q8EA37, Q8FJK5, Q8P7P7, Q8PJ10, Q8Z861, Q9CKN9, Q9I541, Q9KNW0, A6V6X8, Q02KY4, P44743,
Q6MGT1, B1ZW93, B2ULZ9, B3DYX1, Q254N4, Q5L5W7, Q6MBU9, Q7UK39, Q823A0, Q7UPG1, B0SGD8, B0SPT9, Q04R21, Q04ZD0, Q72PC8, Q8F710, A0LV11,
A0Q0P6, A1SJ39, A2BNP5, A2BU62, A2C661, A3DDZ7, A3PAG5, A4FLZ0, A4J5U4, A4XLD9, A5D2R3, A5GNW7, A5GQP4, A5I4I1, A5N854, A5UQQ2, A6LSR6,
A6TRJ4, A6W833, A7FVY1, A7GFZ4, A7NIS8, A8G2A6, A8MLX7, A9AZS3, A9BCV9, A9KLS2, A9WDA3, B0CB83, B0K1C1, B0K9N5, B0TIH8, B1I310, B1II37,
B1KWJ5, B1VXY2, B1WUD1, B1XPZ7, B2A3C0, B2IVR7, B2TJ67, B2V4I1, B7JV66, B7KDB6, O86812, Q0AXI3, Q0I735, Q0SSE5, Q0TPS8, Q10Y85, Q18BJ2,
Q1J1F6, Q24W37, Q2RJK1, Q3ACX5, Q3AH63, Q3B002, Q3MFH1, Q47RT4, Q55803, Q5SJ39, Q67NX5, Q6A908, Q72JG1, Q7NDB8, Q7U477, Q7V3H3, Q7V8Z5,
Q7VE92, Q82K95, Q895I7, Q8DIL8, Q8RA52, Q8XJS9, Q8YXJ1, Q935Y2, Q97I40, Q97L63, Q55914, A5IL80, A6LM59, A7HMK2, A8F8W3, A9BEU9, B1LAG0,
B7ID25, P39409, Q01QF9, B5YF42, B8E0X3, B1H070, A0LY94, A1BGN4, A4SEQ5, A5FJ06, A6H1B8, A6L3I9, A6L8G0, B2RMI0, B3ED49, B3EJF5, B3QS43,
B4S808, B4SA62, C1A949, Q11QF0, Q2S0P9, Q3ASD4, Q3B4B8, Q5LGK5, Q64XE8, Q7MTB0, Q89ZK5, Q8KD71, Q8RFZ9, A3PFQ4, B3Q9D7, B5ZTF1, Q07VN5,
Q13D92, Q21DC2, Q2J405, Q2RP22, Q92L68, A0LBZ1, A1B4Z8, A1UUF7, A4WNI9, A4YJY2, A5E8P3, A5VAJ8, A5VN22, A6UE14, A6WV17, A7HSW7, A7ICB3,
A8IQ73, A9IL44, A9M6S9, A9W383, B0CII9, B0T387, B0UQR1, B1ZG98, B2IGZ5, B2S7X6, B3PQY8, B4RCA4, B8EIR0, B8GXM4, B9JCI9, B9JU97, B9KQP1,
C0RGD9, C3MAJ1, Q0ATR3, Q0BV08, Q0BWY9, Q11EE0, Q16DM2, Q1GC70, Q1GV98, Q1MAN2, Q1QHE3, Q2G8E3, Q2N9J2, Q2YNV3, Q3IY22, Q57FT9, Q5FUA9,
Q5LN66, Q5NNQ4, Q6G1C0, Q6G592, Q89X03, Q8G374, Q8YEL1, Q98E86, Q9ABT6, A0K7T8, A1ISB3, A1K3Y6, A1KUD6, A1TM24, A1V4K3, A1VNF1, A1WE19,
A2S2A0, A2SHB8, A3MK77, A3NA56, A3NVX3, A4G4J9, A4JEP2, A4SYE2, A6SZX3, A9IK57, A9LZN6, B1JT94, B1XXL6, B1YR46, B2JIV3, B2SXT2, B2U9U6,
B4EAX1, B4RMG2, C1DD41, Q0AE39, Q0BEW5, Q0K959, Q12AB5, Q13X26, Q1BGX6, Q1H0U6, Q1LLI8, Q21W25, Q2KY87, Q2SWE6, Q2Y6F3, Q39FQ7, Q3JRQ1,
Q3SL73, Q46ZI0, Q47BR3, Q5F911, Q5P7B0, Q62JW2, Q63UT5, Q7NS85, Q7VWK8, Q7W6P5, Q7WHM8, Q8Y032, Q9JZ42, Q47GW8, A0LQM1, A0RRU0, A1AL40,
A1VAL8, A1W1W6, A5GEC2, A6Q115, A7GVW3, A7H662, A7HCD6, A7I414, A7ZGB0, A8EQW8, A8FP27, A8ZV25, A9FFJ6, B2UVG6, B5Z947, B6JNS0, B8DRU2,
B9KEA4, B9L721, C4XTP4, O25970, Q0P7R8, Q17ZF6, Q1CRK2, Q1D6I6, Q1DCU1, Q1MQJ3, Q2IIC5, Q2LUM5, Q30NR4, Q30X35, Q39S71, Q3A2Z4, Q5HS83,
Q6ALW1, Q727F1, Q74E53, Q7MSW1, Q7VGY9, Q9ZJI4, A1JKR9, A4TMU0, A5CX33, A5F3F8, A7FFY6, A7MU39, A9R805, B0RT51, B0U494, B1JS02, B2I7V5,
B2K9Q3, B7VJT5, C3LT10, Q1C5I5, Q1CK94, Q2P2T0, Q4UUL5, Q5GZT3, Q667Z6, Q7CJM9, Q7MNF3, Q87B36, Q87S19, Q8D1Y5, Q8DEZ6, Q8P984, Q8PKZ1,
Q9PG43, A0KJ41, A0KUJ2, A0Q6H0, A1AW44, A1S866, A1SU36, A1TZP7, A1WXZ3, A3D6W2, A3M208, A3N1S4, A3QCF9, A4IY03, A4SP04, A4WD95, A4XY35,
A4Y8U3, A5EVN1, A5UAC2, A5VYT2, A5WGQ4, A6TCD6, A6V0V7, A6VQX9, A6VV03, A6WQQ0, A7MGV3, A7NC58, A8A323, A8AD69, A8FT67, A8GHW8, A8H242,
A9KFV0, A9KXL1, A9MHL3, A9N1Z8, A9NDW2, B0BQK6, B0KPI4, B0TLI1, B0U083, B0UWR0, B0V4U0, B0VKS2, B1JDQ5, B1KKI9, B2FNQ6, B2I3E2, B2SGH6,
B2VE98, B4EZT6, B4SSW3, B4T0Q1, B4TD95, B4TR97, B5BAY3, B5F1A0, B5FAW9, B5FR66, B5R584, B5RCZ4, B5XNL2, B5Z0Y6, B6EGY4, B6IZM7, B6J7Q9,
B7H072, B7I5G4, B7LKC1, B7N6A5, B8E9S4, C0PYM8, C3K1L7, C4LC34, C5BET4, P44665, P57373, Q057Q1, Q085U9, Q0A989, Q0BLY6, Q0HKW2, Q0HX60,
Q0T202, Q0VND7, Q12PT7, Q14HF5, Q1IEI4, Q1QD22, Q1QTL1, Q21KT6, Q2A3H3, Q2SDW1, Q31I07, Q3ID16, Q3JCN4, Q3K7B3, Q47WB7, Q492D9, Q4FTX0,
Q4K6U6, Q4QNH7, Q57LI4, Q5E775, Q5NG03, Q5PNI4, Q5QYC0, Q603C0, Q65R87, Q6D273, Q6FEM6, Q6LU52, Q7N709, Q7VNZ4, Q83C77, Q83K42, Q88PK0,
Q89AK8, Q8EC29, Q8FF55, Q8K9P5, Q8Z4P2, Q8ZN52, Q9CJJ8, Q6MPV7, B1ZQZ5, B1ZVM5, B2A0B9, B2UNF2, Q6MDD0, Q6MED6, Q7UHU7, B0SGA8, B0SPQ8,
B2S216, O83107, Q72NP7, Q73KZ3, Q8F7V1, A0AFT6, A0LV48, A0PQ89, A0Q112, A0QVE4, A0RHN7, A1SLQ4, A1T787, A1UEJ3, A2BT57, A2BYK7, A2C4M8,
A2C6T3, A2RDK1, A2RHR3, A3CLL3, A3DCX9, A3PEX0, A3PXZ7, A4FMC5, A4J582, A4QF26, A4TC75, A4W3A5, A4XL78, A5D1B6, A5GNG9, A5GVE8, A5I4T4,
A5ISA3, A5UT23, A6LSK1, A6QGB8, A6TRW3, A6U137, A6W7W9, A7FW72, A7GG92, A7GRJ4, A7NPY6, A7X1H8, A7Z4J5, A8AVZ7, A8FD40, A8G6Y2, A8L6D8,
A8M6B6, A8MH89, A8Z3Q4, A9AY55, A9BCH2, A9KM95, A9NEU7, A9VTA2, A9WFY6, B0C9F4, B0JT33, B0K1Y9, B0KA06, B0S143, B0TGT1, B1HQE6, B1I501,
B1IAU3, B1IIL7, B1KX56, B1VYT2, B1WU13, B1XQH8, B1YG36, B2A2K6, B2GJ15, B2HJP3, B2INF0, B2J6D0, B2THS9, B2V4B5, B4U1T1, B5E369, B5XMB1,
B7HDY7, B7HLJ7, B7IUM3, B7JJV1, B7K4N4, B8DCJ5, B8FS78, B8I259, B9DPM7, B9DUW7, B9IVF3, C0MBZ4, C0MD67, C1B2V0, C1C6B0, C1CJL5, C1CVX7,
C1EP88, C1KZZ3, C3L763, C3P635, C4Z523, O34617, O86754, P0DF10, P0DF11, Q032R6, Q03J17, Q04LD5, Q0AXL8, Q0I7M1, Q0SS81, Q0TPL4, Q110I1,
Q182S0, Q1AYW0, Q1BAG9, Q1J0P9, Q1J5R7, Q1JAS5, Q1JG02, Q1JKX6, Q24U12, Q2FHM0, Q2FZ66, Q2J713, Q2JMN2, Q2JRQ8, Q2RK16, Q2YXJ8, Q318R1,
Q31MD1, Q3AC22, Q3AHX9, Q3AZA0, Q3K2R2, Q3M9B9, Q46J26, Q47S46, Q48SK0, Q49WZ9, Q4JV24, Q4L5R9, Q55880, Q5HGL4, Q5HPX3, Q5L0S1, Q5LY98,
Q5M2V3, Q5MZJ6, Q5SGZ3, Q5WKB6, Q5XAZ6, Q5YS67, Q636G2, Q67PQ7, Q6G9Z5, Q6GHL7, Q6HEV1, Q6NGK9, Q6XK03, Q723G9, Q72HC1, Q732K6, Q73VR8,
Q7A600, Q7NIV3, Q7V010, Q7V5P5, Q7VA32, Q819U3, Q81WH4, Q82JY3, Q833B6, Q895P8, Q8CSW0, Q8DG98, Q8DQG7, Q8DVG8, Q8E1A3, Q8E6Q7, Q8ELW7,
Q8EVK0, Q8FP78, Q8G481, Q8NP06, Q8NX16, Q8P058, Q8R9T4, Q8XJL6, Q8Y9P2, Q92EH6, Q97IC4, Q97RN5, Q99UQ0, Q99YU5, Q9CJ27, Q9K9Y8, Q9RVT6,
A5ILF6, A6LN47, A7HNQ1, A8F8C2, A9BIC0, B1LAW7, B9K7Z6, Q9X240, B1Y6D6, Q2L1Z5, Q3SHI2, Q7NVT9, A9FD89, A4VLN9, A4XU99, Q9I2Q6, A1A917,
B7MGN3, B7MQM8, Q1REF5, Q8FJQ3, P30140, Q9S498, Q9K7C9, B5QWC1, B5R817, Q5HKJ7, A0L887, Q72DS4, Q4K5I3, A7ZJJ0, B1LM72, B5YRM0, B6I7T6,
B7LC63, B7M753, B7MQN5, P65382, P65383, Q324B1, B0SGV6, B0SQ03, Q04UG1, Q04Z14, Q4K4T2, A1W574, Q72DE5, A6QCU6, B0RTE0, A6T6T1, A1KMM4,
C1AFZ5, P0A645, P9WH14, Q1IRD1, A1B656, A7HBU9, Q4URN0, Q8PBX1, P64554, P64556, Q484J7, A1AE55, A5IC42, A7ZPW0, B1IWE4, B1LNH2, B1XAZ2,
B2TXU2, B6I589, B7LDA9, B7M7M1, B7MI02, B7N304, B7NRG7, B7UGW3, C4ZX92, P36979, Q0TEW8, Q1R8L6, Q31XX3, Q32D45, Q3YZ35, Q73JG6, Q0TTH1,
A1KKR0, A5U4P6, C1AQD3, P65284, P9WK90, P9WK91, Q8DLC2, P9WJS3, O31677, Q9X2H6, B8GW82, Q9A7I8, Q4FNN5, A0LIM0, B2FUS8, B4T0B2, B4TQZ6,
B5BBX8, B5F0X7, B5FPB9, B5QXV6, Q57RA8, Q8ZQM1, Q5WWH4, Q5HKI0, Q8CTX5, Q0RDQ8, B9KBR9, Q7WB85, Q7WMQ3, A6T6L5, Q0T6I5, Q83S46, Q93GG2,
B7JLW9, Q635G7, Q6HE51, Q81MB3, A0A069AMK2, Q7U8K0, O66732, Q8KBK9, A1IRY3, B4RLI6, Q5F8G2, Q9JRW7, Q9JZA5, Q02SE8, Q187U6, A0LVG0,
Q0S277, A9MUH8, B4SYN0, B4TB74, B4TPZ0, B5EZB2, B5FNB2, Q57RQ7, Q8Z8G5, Q9RCI2, B0UUK6, A3M659, B0VMU3, B2I330, Q48LZ7, Q4ZX26, Q886Z3,
Q5X516, Q5ZV93, A4IM49, P09825, A4TNY6, Q02RW0, B0JJS5, A1A280, Q48FA7, Q87Y01, A5G209, P0A1E1, P0A1E2, Q2GJJ5, P32675, A5IRA1, A6QFD6,
A6U030, A7X0C7, A8Z1I3, P65285, P65286, P65287, Q2FIE9, Q2FZX4, Q4L4T8, Q5HHG0, Q6GB01, Q6GIG3, Q6A7K4, B5Z928, Q1CRM5, Q48DM6, Q4ZN84,
B4RLK9, Q5F8I0, Q8Y2L3, Q9JUC8, Q48HV8, Q9KTX3, B9J9I5, Q0C606, Q8ZGW5, Q48CS1, Q88A98, P75794, Q2NS37, Q2J5A7, Q65JS3, A5VUG1, A9MBD6,
A9WYH2, B2SBF0, C0RL26, Q2YKB8, Q577P8, Q8FWG3, Q8YBW1, A1V817, A2S7R3, A3MNG1, A3N520, A3NQS1, Q2T1Q4, Q62MW9, Q63Y25, A7ZY37, B1X7B1,
B4SZK4, B4TQU6, B5BC24, B5F079, B5FP68, B5QX73, B5R769, B7MGN9, B7NA81, B7NNL1, C0PWZ0, C4ZXV3, P30745, P65386, P65387, Q32I50, Q3Z403,
Q5PG37, Q83S40, A2AXI2, A5HBL2, Q9FBG4, Q0SS32, A3M4U4, B0VCA8, B2HYX9, B7H3S4, B7I4I4, A1A1I4, Q6A7V7, Q5PAD4, Q7VVF1, A6T689, B5XZS6,
A6GZL9, Q5KVM7, A0RME8, A6L1U8, Q18CP3, P0A9N4, P0A9N5, P0A9N6, P0A9N7, B7JDM3, C3LCA6, C3PDK2, Q6HBT4, Q81XM8
```

Aldehyde dehydrogenase (PF00171), referred to as ADH in Chapter 4:

O67166, Q3B2U3, B0T8I8, Q1GV29, Q2G9T9, Q9A7W2, Q9AAL5, P54222, Q92YD2, A9IMB2, B0T315, Q1GHC8, Q3SVI0, Q8UBS1, Q8YJ78, A0K608, A1V675,
A3NBM7, A3NXG2, A4JCW0, A9AD19, B1JYT5, B1YW19, B4ECZ8, Q0BGV0, Q141D3, Q1BXP1, Q39I25, Q3JQC6, Q62LN5, Q63SD7, Q7NXX7, A0K4I3, A1WGI4,
A4SVE2, A6T242, Q1BZ67, Q39JM2, Q3SG61, A9EN94, P53000, A1AB46, A4WAR9, A7ZLN7, A8A002, A8AGJ9, A8GHZ8, A9MQY3, B1IS15, B1LFH3, B1XDF5,
B6IAI9, B7L6F9, B7LR95, B7LZ33, B7MMS8, B7MUN0, B7N4K1, B7URJ0, C4ZVI3, C6DD82, P77674, Q0T431, Q0THX6, Q1RBX3, Q32FQ5, Q3Z1H6, Q6D6Y7,
Q83R90, Q8FHK7, H2IFE7, P12693, P25553, P33008, A0KN18, A0KST2, A1RNU4, A1U5W8, A3D039, A4SK35, A4W9J7, A4XWE7, A4Y344, A5F529, A5ICK3,
A5W0D5, A6T7T5, A6WST7, A7MNV8, A7MX96, A8A0T8, A8AHD8, A8FRE9, A8GFQ3, A8H8E0, A9L3U4, A9MFG0, A9N276, B0KR47, B0TU38, B1JCH3, B1KDF2,
B1LDY5, B2U3C8, B2VJX8, B4T3Z6, B4TGE2, B4TUC3, B5BA70, B5F7J0, B5FBM2, B5FJD6, B5QWI8, B5RAZ9, B5YQ33, B6EML5, B7L6M0, B7LQ46, B7MVM5,
B7N582, B7NT31, B7USC7, B7VLI4, B8CIT6, B8EBC2, C0Q6X8, C3JXY7, C3LRS7, O50174, Q07XY1, Q0T4V3, Q0TH84, Q12JA9, Q12QD2, Q15Y60, Q1I6Z5,
Q1QTQ7, Q2SKP1, Q321N8, Q32G88, Q3IC91, Q3IFT7, Q3K885, Q488Y0, Q5E2G9, Q5PHC0, Q5QVX3, Q5ZUT5, Q6FCQ0, Q6LVE5, Q7ADE6, Q7MH21, Q7N2G9,
Q7UCI7, Q87L22, Q88EI4, Q8DCS9, Q8EJ54, Q8FH01, Q8Z6G1, Q8ZPV0, Q9KNW4, P17445, Q8FKI8, E1V7V8, Q88RC0, Q9I6M5, P80668, P0A391, B0U111,
P07004, Q21FC9, Q31IE5, Q4FUZ5, Q5E6W0, Q5GZF2, Q9PEM3, P23883, O06478, O34660, P39616, P42329, P46329, P94358, Q2FK94, Q2G1J0, Q2YUN1,
Q2YV11, Q49Z69, Q4L803, Q4L919, Q5HJK3, Q5HLA3, Q6G7I8, Q6GCV9, Q6GEV3, Q6GKD8, Q8CN24, Q8GAK7, Q8NVG4, P0DPF0, Q59931, P9WNX8, A0REB5,
A7BJC4, P42412, Q5KYK0, A2RJM8, A5GSH0, B7GTL2, P39821, P54903, Q1IZP1, Q3AYD4, Q3MH53, Q47MW1, P38947, Q9A777, A3PI00, A4WUY6, A6U6Y9,
A6X2G8, A9M9H7, B0CKN3, B3PTE1, B5ZUG3, B9JBA3, B9KNS6, C0RHQ3, C3MIE5, A0B2F6, A1UVS4, A2RWD6, A3MEC6, A3NKP8, A4JJG5, A9AN00, B1K708,
B1Z033, B2JS88, B2TCJ9, B4EHJ1, A3M365, A4TNP1, A4VKC2, A4XPI6, A5WA96, A6VEI4, A6W2P7, A7FKL5, A7N2Q0, A7ZI51, A7ZWV5, A8GBX8, B0KN18,
B0RNV0, B0V944, B0VST2, B1J0W5, B1J2K9, B1JSQ9, B1LIJ8, B1XET7, B2FQ90, B2HV80, B2K8U5, B4SHW0, B5Y007, B5Z1R1, B6I075, B7GYG4, B7I896,
B7L441, B7M2V6, B7N8L4, B7NK50, B7UJG5, B7V5R4, B7VQ28, C6DKY5, Q9KWS5, O86447, Q9I6C8, Q1MJU3, Q2KB42, Q3J4E9, Q8G1Z9, Q985M6, P55653,
Q0B712, Q13NG6, Q1BQE1, Q39A43, Q62CH7, Q63KK8, H1ZV37, Q02DY9, Q0T7M9, Q0TKW0, Q1C931, Q1CFR8, Q1IG69, Q3BXK7, Q3K5H4, Q4K4K8, Q4UYN4,
Q66D53, Q6D6E0, Q6FDF8, Q7AH91, Q7MF13, Q87H52, Q88CW7, Q8D3K3, Q8P5D8, Q8PPG7, Q8ZGV9, Q9HTJ1, Q9I702, Q9L4K1, P25526, P86808, P19059,
Q2FWX9, Q3C1A6, A0PN13, A0QMB9, A1KF54, A5TYV9, O32507, O69497, P9WNX9, Q55585, Q73TP5, Q7U2I0, Q1JUP4, P28810, A0RDW1, A4IPF5, A5YBJ3,
A7GPH3, A7ZAI1, A8FDV4, A9VF06, B7H597, B7HR31, B7IW48, B8DCT8, B9IZZ7, C1KZ99, Q4V1F6, Q5L025, Q5WH11, Q5WKZ1, Q63B74, Q63BL0, Q65IX1,
Q6HJ19, Q738L2, Q81DR5, Q81QR5, Q8EMV4, Q8ES27, Q8Y9Y4, Q92EQ7, Q9KAH5, P42269, Q84DC3, P0A390, Q8YV15, Q3JNN5, Q63QT9, P94682, Q9AHG1,
Q59702, Q6F9G0, Q79EM7, Q8GAI8, B2V9F3, B4U8A0, C0QS00, Q1IS80, B5YEQ8, B8E0D2, B3EE24, B3QPW0, Q2S354, B5YK66, A0LCZ1, A1BAM3, A3PQJ2,
A4YKF1, A5E961, A5FYS4, A5V8T0, A5VSI3, A6UDV8, A6WXS2, A7HT65, A8LK12, A9M880, A9W6Q2, A9WWW7, B0UMS9, B1M695, B1ZFJ0, B2IE42, B3PRZ5,
B3Q733, B5ZUE5, B6JD19, B7KS47, B8EQH0, B8IEM1, B9JER3, B9JUH7, B9KW06, C0RF92, Q07V09, Q0AL88, Q0BWP1, Q11CP8, Q13DN0, Q165Y8, Q1GVM0,
Q1MA74, Q1QQT7, Q21D00, Q28Q10, Q2GCB4, Q2J3J6, Q2K2X4, Q2N9V1, Q2RV06, Q2VZT9, Q3IXX7, Q5LRY6, Q5NLX5, Q6NDE4, Q89X85, Q8FYM3, Q92LB2,
Q98EZ5, Q0K845, A1KTV4, A1VTR9, A1WCP4, A2SCE2, A4G8E9, A4JBI4, A9C1G5, A9LYY9, B1YTB0, B2JGX3, B2SYN3, B2UC98, B3R6R0, B4RLE6, B9MH63,
C1D6E4, Q0AEA6, Q0BIB2, Q0K710, Q122S5, Q13U85, Q1GZB2, Q1LJ33, Q220P2, Q2KXE0, Q2YBP9, Q46XE1, Q47IN4, Q5F8D3, Q5P255, Q7NQ51, Q7VWZ0,
Q7W9M7, Q7WH30, Q820I7, Q8XVT6, Q9JUK8, Q9JZG3, A0LPG2, A1AUT0, A1VCR9, A1VYR7, A5G906, A6Q3B4, A6QB48, A7GVZ7, A7H6U9, A8EVN0, A8ZRY3,
A9GVS8, B3E3M7, B4UM59, B5EEI4, B8FM70, B8J495, B8J8Z7, B9L9A2, B9M0D6, C0QLF1, C4XSQ4, C6BSC1, C6E7L9, Q2IMG0, Q2LU85, Q30SG0, Q311G6,
Q39QR2, Q3A1E0, Q6AK09, Q72AN9, Q747Q4, Q7M8Z4, Q7VI05, A0KNQ8, A1A7V4, A1AVH0, A1JNX6, A1S8L1, A1SYT9, A1U3C3, A1WYZ4, A3M1Z8, A3N3P3,
A4SJF6, A4TPK0, A4VR07, A4W6X5, A4XYY4, A5CXP4, A5UBQ2, A5UF66, A5W9J6, A5WH78, A6T561, A6V0A3, A6VMV2, A6VZ85, A7FLI1, A7MI49, A7ZI02,
A7ZWK6, A8AKP4, A8FYU6, A8H764, A9MNR4, A9MY02, A9R2X0, B0BTX9, B0KJY5, B0RS59, B0TQC4, B0U208, B0USH2, B0V4S6, B0VKT1, B1J0Z1, B1J133,
B1JIH2, B1KD27, B1LHT9, B1XDY6, B2I3D3, B2I7L2, B2K6Q4, B2SHW6, B2U3T4, B2VHM6, B3H321, B4EUV1, B4SVW6, B4T7Q6, B4TZ91, B5BDP7, B5EWK6,
B5FBJ5, B5FJX5, B5R4S6, B5R5R9, B5Y169, B5Z1I7, B6I023, B7H081, B7I5F5, B7L3Z5, B7LNF7, B7M272, B7MC93, B7MQ79, B7N8H4, B7NK82, B7UJD1,
B7V8A7, B7VJB0, B8CKF0, B8F6K8, C0Q6U2, C1DMQ0, C3K2M9, C4LBK4, C4ZTA0, C5BNN0, C6DCX4, P45121, Q02SH4, Q0ABN0, Q0I2E5, Q0T7R6, Q0TL73,
Q0VN49, Q12SX8, Q1C4E7, Q1CLC6, Q1I4F0, Q1QDZ7, Q1QXB4, Q1RFS7, Q2NVE9, Q2P2G1, Q2SA27, Q325P3, Q32J27, Q3BSJ1, Q3IEY7, Q3J7T1, Q3K693,
Q3Z594, Q47UQ0, Q48DL4, Q4QJW6, Q4UVI0, Q4ZN73, Q57ST2, Q5PF69, Q5QY68, Q606Y1, Q65S49, Q66DY8, Q6D1I4, Q6FEN5, Q6LTX2, Q7MN58, Q7N7B1,
Q83SH9, Q87EK9, Q87RU9, Q87VV6, Q88DL4, Q8DF94, Q8EHU1, Q8FKM3, Q8P8K3, Q8PK35, Q8X7N4, Q8Z932, Q8ZC09, Q9CM98, Q9HX20, H8ZPX2, Q2BN77,
P0DOV9, O05619, O69763, B1ZMC1, Q7UNV2, B0S9A5, B0SRT9, B2S2U7, C0R0B8, P74935, P94872, Q04Q92, Q054P8, Q72NQ9, A0AI64, A0PU04, A0PXA4,
A0R115, A1KLC0, A1SHP5, A1TC11, A1UIZ7, A2BQ71, A2BVQ3, A2C148, A2CAS7, A2RD38, A3CMT1, A3DC22, A3PBW4, A3Q2E3, A4F9M1, A4IPN4, A4J3Q0,
A4QG75, A4T2I0, A4VTT5, A4W028, A4X1X0, A4XK60, A5CR34, A5CZ28, A5FQ48, A5GJS5, A5N0V1, A5U5C2, A5URC6, A5VIE0, A6LPD5, A6TUA0, A6WDM6,
A7NSB7, A7Z3T1, A8AX77, A8G3V6, A8L1V4, A8M064, A8MFQ5, A9KMV0, A9VK31, A9WBM7, B0CFL0, B0JWW5, B0K0T2, B0K9C5, B0TBV8, B1IBA0, B1MX68,
B1VXE5, B1XLA4, B2G5W8, B2HME9, B2IP88, B2IZ89, B2THG5, B2UX78, B4U4K6, B5E435, B5XML5, B7GJH1, B7H673, B7HVK6, B7ILK1, B7JD15, B8D0Y6,
B8DHP3, B8FUB6, B8G8E0, B8HYG3, B8I6T0, B8ZP35, B8ZRN0, B9DV79, B9E4Q5, B9J498, B9LE47, B9MK80, C1AQZ3, C1C6R4, C1CDS9, C1CK18, C1CRW1,
C1EZ15, C1L2G7, C4Z075, C4Z9V4, C5D2V2, O86053, P0C1E0, P0C1E1, P0DD20, P0DD21, P54902, P65789, P96489, P9WHV0, P9WHV1, Q02XW0, Q035M3,
Q03J03, Q03ZF1, Q04FB2, Q04KZ2, Q0AWJ6, Q0I8Z0, Q0SH62, Q0SPX8, Q0TM73, Q112S1, Q1B639, Q1J5F6, Q1JAG3, Q1JKL4, Q1WRR6, Q2JDN7, Q2JN71,
Q2JQB4, Q2RKZ6, Q31BU4, Q31KX4, Q3AF39, Q3K395, Q3Z6Z9, Q3ZYH9, Q46LW0, Q48RY7, Q55167, Q5KYA2, Q5LY84, Q5M2U1, Q5N0Z7, Q5SH02, Q5WH54,
Q5XAL0, Q5Z025, Q639W9, Q65IS9, Q65KU7, Q67LC2, Q6A9H6, Q6AFX9, Q6HHC2, Q6NFW0, Q735X3, Q73XR2, Q7NEF6, Q7U654, Q7V293, Q7V8C3, Q7VBM1,
Q82C81, Q839W3, Q890J4, Q896G4, Q8CUQ4, Q8DKU1, Q8DQ60, Q8DVM9, Q8E1R9, Q8E783, Q8FN87, Q8NZX9, Q8RAE5, Q8XHA7, Q92CE5, Q93Q55, Q97E62,
Q97R94, Q99YJ8, Q9CBZ7, Q9CF73, Q9KCR5, Q9RDK1, Q9RTD9, A5IKB6, A9BJ18, B1L9J9, C5CE09, P23105, P43503, Q5HMA0, Q8CNI5, B3WA82, B1JVH2,
A1A1U9, A1URN6, Q4K5F9, Q6G0S6, Q6G4Z0, A1KAH8, A9HWX2, A5VPA5, B2SA42, Q2YMP8, Q57EI0, Q8YFY0, A3P6B0, Q3JLL8, A1TVU4, A8GAD4, Q8UH56,
Q92UV7, Q2SXN9, B5Z0W0, Q8X9W5, Q5WVZ4, P40861, Q2FF06, Q2FWD6, Q5HE78, Q7A1Y7, Q7A4D8, Q7A825, Q99SD6, Q99X54, Q3AKU8, Q9WYC9, B6IBG6,
Q48G19, Q4ZQH8, Q6F9F7, Q4K837, Q885J7, P42236, P94428, P76149, Q8G5H9, P17857, Q65D00, Q24XR6, Q720G3, Q6HIK3, Q81QB6, B7MCD1, Q2SZ88,
Q62H23, Q5X4K4, A4IPB2, Q5KYR4, Q88AE9, A9VMS6, Q723T1, C3K3D2, Q48CM6, Q4ZM62, P71016, A7ZML4, B1IPI4, B1XGK7, B7M1F8, B7MAV7, C4ZZA2,
P76217, Q1RB47, Q4JWT3, Q2YLI7, Q57B47, A5F602, Q9KPT9, Q9A2X6, A9MYQ4, C0Q4N4, Q57P61, Q5PHV8, Q8Z747, Q8ZPC9, C3LEW9, C3NZU4, Q81P27,
Q5FRT2

Aldo/keto reductase (PF00248), referred to as AKR59 in Chapter 5:

Clade1:
P74308, P06632, P80874, Q46857, P46336, P77256, Q52472, P30863,
Q46851, Q02198, Q76KC2, P77735, P15339, O32210, Q8ZI40, Q8ZM06,
P0A9T4, P25906, Q8XBT6, P58744, Q8ZH36, Q8ZRM7, Q8X7Z7, Q8Z988,
P9WQA5, P76187, A0QV10, O05408, O34678, Q8X529, P46905, P9WQA7,
Q7TXI6, A0QJ99, A0QV09, P42972, O69462, A1T726, Q73SC5, A4TE41,
A1UEC6, A5U6Y1, A1UEC5, A0QL30, A0PQ11, A3PXT0, Q1BAN7, B2HIJ9,
P9WQA4, B8ZS00, A3PXS9, A1KMW6, Q73VK6, P54569, P9WQA6, P63485,
P76234, P0A9T5, Q01333

Clade2:
P80874, P46336, P77256, Q52472, Q46851, Q76KC2, P77735, P0A9T4,
P25906, P76187, O05408, Q8X529, P46905, P9WQA7, P42972, P54569,
P9WQA6, P63485, P0A9T5, Q01333

B

Automatic diverse subset selection from enzyme families by solving the maximum diversity problem

In Chapter 4, a tabu search algorithm inspired by Wang and colleagues [103] is implemented. The parameters used for this algorithm are shown here:

Table B.1: Table showing the different tabu search parameters used in this work and their values.

Parameter	**Value**
Tabu list size	12
Tabu tenure length	50 iterations
Exit condition	50 iterations without improvement
Candidate list size	10

C

Viability of autoencoder-generated sequences for artificially increasing enzyme family diversity

In Chapter 5, a neural network inspired is implemented and trained. The parameters used for this algorithm are shown here:

Table C.1: Table showing the different neural network parameters used and their values.

Parameter	Value
Reconstruction loss function	Categorical crossentropy
Discriminant loss function	Binary crossentropy
Reconstruction activation function	Softmax
Discriminant activation function	Sigmoid
Input and hidden layer activation functions	ReLu
Optimiser	Adam
Batch size	16
Epochs	5
Random dataset size per epoch	5000

In Chapter 5, 25 of the 30 SynthAKR30 synthetic enzymes were tested in SDS-PAGE experiments in the lab. Their primary sequences along with their molecular weight in kilodalton are show here. The ”c1” and ”c2” suffixes represent which clade each enzyme belongs to.

```
>2957_seq_c2(36.6kDa)
MSELDVDGIGQVSLIGLGTMFFGSMEWEGGDYYATAAARAIVKRAAALGRTVTDTAYYYGLGKSETILG
EAFGDDLTTEVYASKVFVVAPGPAPNRRRELASARRLQLRRRPLYGQHGPNPVVDDSVTMVGMRLLLDS
GDIGAAGVSRDHLAWWRKAADALGRPVVVVQVFFSHAAPDALDDVVPFAELENRIVIAYSPLAQGLLGG
GYGLEERPGGVLALNPLFGTECLRIIPPLLATLLAIAVDVDAPPAQAVLAKLSQLPQVVDIPGSSSVPQ
LEDFIAAADTERSASSQDALTAAALAPRPVSTGRFLTDMYREKVSRRQ
```

```
>4466_seq_c2(40.4kDa)
MQRHHIIHSHLETSTLGLLTFMFGMQNSEADAHDQLDYAVDAGGGNIDVAMMYPVPPRPETQGLTVTYV
GRWHYKRGSEEKLIIASKVSSPSRNRKNGIRPDPALDRNNIREALHDHLKRRGTDYLDLHQVHWPSPPT
NCHGKHGHSDTDSAPAVSLLDTDDLLAEYQHAGKIRYIGVSNERTAGVMMMLLLDDKDDLPRIVTINPP
YSLLNLSSEVGLAEVSQCFGVELLAHSQLGGGGLTGYFLRGWGKAPARNNLQSTFTYYSGEEIQQDVAA
DVDIHHRHALDPAQVARAGVQLQPFVASTLLQATMMDQLQTIIESLHLELSEDVLAEIEAVHQVYTYPA
P
```

```
>5050_seq_c2(36.5kDa)
MKYLDVDGIGGVGNIGRGTPQYGLRSWFMGDRFAVGAARDNVDTAAATGVVLDDTAEIYGLGKEERILG
EAEGDDRTEVVVASKKGPVPPEPAVIKNRARADHSLRQLNLRPLYQQHQPHPVPPDSVIMPGMLDLLDS
GDIGAAGRSRHSLYRWAKDDAALGRVVVSNQVHSALAHPDALLLLVPFDEREFLIVIAYASLAQQLLGG
GYGLENRPGGVAALNPLAGREELHIILPLLATLAIIAVDVDPKPAQVALAWLIWLPGVVDIPGASSVEQ
LEFEVHAADNILSAQQNDADTDAPHAFMPVSTGLGLVDEVREKVSR
```

```
>6626_seq_c1(31.9kDa)
MTGTTGFAAQIPSTLLDDFSTPPLGLGGTGSLSDETARRAVAALEEGGQLLDDAAAYGQEEAVARRIAA
AGGVPRELQVVTTKHAVADGGFTSTYADARASLSLGGHDVYYSHLIHWAPGVGFYMMDWGGGRTSHKGG
GAAHIGVCCFFAEHHLNNIIDLFMFPAVIQIQLHHLLNQSEELRCCYQQQVVVTEAYFPLPGGLLDDDA
AVQIAASAHKTPAAVRLLLWLQHHGTVVVRHAAPNAIAISLHMVDFFLLTDDMAATLNLLDGGVRNPDD
PETAGG
```

```
>8092_seq_c1(32.4kDa)
MTGESGAAAAPSITLNDEAEEPVLGLVVAELSDDETERAVAAHLEIGCRLIDTAYYYGNEEVVGRAIAA
SGVARAELFVTVKLATPDGGFTRSQEACHASLDRLGLDYVDLHLIHWWAPPVGCYVDAWGGMISSNGEG
HARSIGVSCFEAELNENLIDLTFVVPAVIIIELAPLLGQDELCDKNAQHVVVQQSYCPLLLGLLLDNDT
VTSIASEYVKTPAQVLLLWRLQLGAAVVVRDARPELIASAFDVFDGELAAMAMDALGGLGLGTLVRDDP
LTYAGQEDP
```

```
>9223_seq_c2(38.3kDa)
MRKKKLGTSDLDISEVGLGCMGHGTEKFKARIISDADIARGIIYLDTADLYDRFRREEIVGDQINNRRI
DINLATKAGWRWDDGEEGWYMDPSWAKIKEAVSSSLTLLKTDIIDHYQLHGGVIEDNDDETIEAFELLK
GGGVIRIQGIISIHVRVIKEYYSKINIVSIMVMFSHFDRDPHWWLPLLEEAQISVVAQGPVAKGLLTEK
PLDQASESMWQNGSLSHSGEEHRNANWAMEAVAPDLSMTEKSLQYLLDQRAVASVSVGAAIIEQRTRTE
QADNARLLTEEEIAQLFSHTKQDKEKAHLSRDPVMEERPPHALE
```

```
>9261_seq_c1(32.7kDa)
MTGPRGPHADIPSVSLNDGSTPPVPGHGVGFLSESAAERSVHAALEAGYRLIDTAAGYGNEAAVGRAAA
AIGTPRREIYVTKKHAPAAQGEQTSSDAARASLERLGLDYVDPYHIHWPAGDAFYYCDYWGGLMQYDQD
GVDRSQGVCNFEAEHLSIIITLSFFAPARNQIEEHPLLNQAHLREVNYQYGNVTEYYGPLGVGTLLDHP
AVTGTAQAHADTPAGVLRRWSIQLGVVVIIRSANPARITSNLEVFDFFLTTDEEATRLGLDFGTRFRFD
PATYTGP
```

```
>9840_seq_c2(39.7kDa)
MVWLAIPERCGTMFYRFCGKSGLQLPALLLGWWNIHGQVNALLSQLHRLIKALDLGITEFDLAPIYGPP
PGSAEHEQGRLLIEEGAAANDELIISTKAGMDMFWGPYGSGGSKKYLLAALDLSLKLMGLEYVDIFASH
SVDECTPMEETASALAHQVTSGKAHHVGISAASPERTQKEVELLQEKDIPHLPHSPSYHLLIRWVDKYG
LLLTLQNGGCGCGDFHPLAGGLLTGKYHPGIPADSVVHQEGNKVNGRTKWMETEAALWLLQLRNEEAQQ
RGQSMAEMALSPLLKLHETTSVLVGASAALQLEESVAALNILTFSTEELAQFDQHIADGELLLNGASSD
K
```

```
>12268_seq_c1(32.5kDa)
MTGFAGAAPAPSITLNDEHTMPVLGLGVGELSDDETERAVAAALEIGCILIDTAYAYGNEAAVGNAIRA
SGVDREELFVTTKLATPDGGFTRSQEACRASLDRLGLDEVDLHHIHPYAPPVGKYVDAWGGMIQPRGEG
AARSIGVSCFTAEHIENLIDLTFVVPAVIQRELHPLLPQDELRDKNAQHTVVVQSYCCLALNRLLDNPT
VTSIASEYTKTPAQVLLLWRLQLGNAVVVRSARPERIIIAFDVFDFELAHEHMDAAGGLNDGTPVREDP
HTYAGT
```

```
>13192_seq_c2(37.0kDa)
MDYLDVDGIGGVSRIGLGTWQYGSREWGMGDYYAVGAARDICKLALALGVTLTDTAYQYGLGKSEEQLG
EALGDDRTEVVSASKVFVVPPEPAVIKNRMRASARRLQLNLRPLYQQHIPHPVVPDSVQPPGMRDLLDS
GDIGAAGRSRYYLARCRKADAALGRPVTGNQVHHHLAHPAALEERVPFDELERLNVIAYSPLHQQLLGG
GYGLENRPGGVRALNPLDGTENLLIIEPPLATLAVIAVDVDPKPAQVSRAWLIWLPGVVDIPGSASVEQ
LEFEVADADIELSAQARDALTDAPRAGRPVSTGLHLVSMVREKVSRR
```

```
>16119_seq_c1(32.4kDa)
MTGSTGAAAAPSITLLDEHTVPVLGGGVAELSEDETELAVLAALEIGCRLIDTAAYYGNEAAVARAIAA
SGRPLHELFVTTKLAPVDQGFTTSQKACNDSLDRLGPDYVDLHHIHWWAPPVGYYVDAWGGMTIRSGEG
HARSIGVQNFTAEEHHIIIDLTFVTPAVNQIELLPLLNQDEHRDKAAQHIVVQQSYTPLVGGRLLDRDT
STSIASEYTKTPAQVLLLWILQLGQVVVVRAYHPERIASHFDVFDGELAEMHMDAALLLGDGTTRPDDP
STYAG
```

```
>23504_seq_c2(39.8kDa)
MVWLDPEERAGCWNCRCCGKSGLLLPHSSSGLWHNFGAVDALESSQAQLIKAGDGAITHGDLANNYGPG
PGSDSEIEGRLLREDEAAARDELIISTKAFMDMMPGPSGSGGSRSSLLAAADSSLKRVGLEYVDIQYSH
SVDEETPMEETASALAAATQSKKALYVGISSYSPEETYKMEELLLEWKIPLLILQPSYNLLRRKVDKGG
LLDTLQNNGGCCIATTPLAQGRLTGGQLQGIPADHAMDMEFVKVTGLTPKWLTEAYLRNRRLLNEMAQQ
LGQSMAFMARSLLLLDDRTTSVLSGARRAIQREENVQALNNLTFSTEELAQIDQAQDDFELNLWQASSD
K
```

```
>23716_seq_c1(32.9kDa)
MANPTIIRLGDGSVMPSLGLGVWQASNEEVIAAIAKALFVGYRNIDTASAYYNEEGVGKALKKASVYVR
ELFTTVKLNNDDQWRPRLALLLSLSKLQLDYPDEYYMHPPVPAIDHDVDAWKGMIALNKEGLCKSIGVC
VFQRHHLQRLIDETGVPPVINQIELHLLPQQLQLHAANATHIQQTESPSSLAGGFGGVDDDDVIRFLAD
KKGKTPAQIVDRWHLDYGLVVIPKSVTPSRIAYNGDVDDFRLEKDDLFEIAKLDAGKRPGPDPDQFGGE
ER
```

```
>23827_seq_c2(36.8kDa)
MKKLAVDGISQVSRGGLQTYQFGSEEWWGGDYYYVGTAIDIVKRARALGVTLTDTAEEYGHGKTERILF
LFLGDDRTEVVAASKVPPVPPGPAVICRRERASARRLTLNLLPLHGQHQPLPVVPDSVPPVGMRDLDDS
GGIGAAGVSRHSLANVRKCDDALGRPSTVNQVLHSLHHPAALEELPDFAEEENNTVIAYSPLAQGLLGG
KYSLENLPGGVRYLPPLDGTEELLPIEPELATLHAIAVDVDAPPAQVALAGIINLPGVPDPPGSSSVEQ
LLFHCADADNELSAQAADALTLAAAAPRPVSTGRFLVSVVRYKVSRSEYR
```

```
>26444_seq_c2(40.1kDa)
MVWLDNPEIFGQMQSRHCGKSGLRLPHSSSGLWANGGHVAAHFSSIAILLKAFDLGITHFDLANNYGPG
PGGDDDNEGVVLLDDDAAEPDPLIIITKAGMDMWPGPGGSGGSRYYLLASSAQSLWNMGLSSVIIQSSH
MMDMNTPMEETAHALAAAVSSGKALFVGISSYSPERPQKMMLLLLEEWWPLLILQPSYPLLRLWVKKSG
LLDTMQNNGVGCARFPLLQGGLLTLKYLLGNPEDSAMAREGMDKRGLTLKMLTEYNHRNSIARRSMAGQ
LGFFMAQVARSLLLLDRRVTSVLSLASRARQLRNERQALNNLEFSTEELAQIDMAPDDEELRLWQASSD
K
```

```
>26820_seq_c2(37.0kDa)
MKYLDVDGDSSVGRIGLGTPQFGSRENEYSYLSAFGAARDNVKTTDALGVTLDDTVEIYGLGKSYLILG
ELHGDDRTEVVVASKYGPVPPFPDVIDKDEHASSSLLQLPRPPLYHWHQNNPVVDDSVIEMGMLDLLDD
GDIGAAGVSNYSLHWWAKADAALGLKVVSNQVFCSLALPDPLLDLVPAAELENRTVIAYSPHGQQLLGG
GYGLEERPGGVRALNPRAGTEELEIILLLAATLRARDVDVDPKPAGAALDWLILLPQGVAIPGASSVEQ
HEFAVARADIELSAQSRSALTDDDHQFRPVSYGTFRTDMVREKESRL
```

```
>32249_seq_c1(32.6kDa)
MEGFSGAPAAPHITLMDEHTTPVLGHGVAELSDDETERAVSAALEIGCNLIDTYYAYGNEFAVGRAIAA
SGVFREELFVTVKKAAADQGFTRSQEACRAQLDRLGLDYVDLHHIHWWAPPVGKYVDAWGGGIQYRGEG
GYNNIGVTCHTAEENNNNIDRTAVVPAVNQRELHPLLNQDEHRKANAQHYNVVQSYCPLAGGRLLDNDT
VVSIASEYVPTPAQVLLLWNLQLGQAVVVRSARPERIISNHDVFDGELAAADMDALGGLGDRTRVEEDP
HTTAG
```

```
>33894_seq_c2(37.4kDa)
MYYLDVGGVGGVSLIGLGTWQYGSWWMMYGYYYATGAARDIVKRAAALGVTLTDTAAQYGLGKSETIRG
EALGDDRTETVVVVKVFVVVPFPAPNRRRELHSRTRLQNRRRPLYQQHQPHPVVPVSVIVVGVRDRLDS
GPIGAAGVVREYLATWRKKAAALGRPVTVNQVHSSHHHPDDLEDLVPFDEHENNIVIYYSPLDQGLLGG
KDGRENRPGGVRHLNPLFGTENLRIIPPTLALLAAIVVDVDPYPAQVVHAWLSWLPQVVDAPGASSVST
LEFNVAAIDIERSAQDADTHTDAALAPRPVSTGLDLTDVVREKVSTR
```

```
>36404_seq_c1(33.6kDa)
MTTFSGAAAAAHITHNDEHYMPVLGHGVFELSDDETENAVSAALEIQCNRIDTYFYYGNEAAVGRAIAA
SGVAREELFVTVKLAAPDQGFTRSQEACRASLDRLGLDYVDRYNIHWWAPPMGKYVDAWGGMIRRRGFG
AARNIGVQNHEAEEHENLIDRTFVVPHHNQEEHHPLLNQDEHRKCNAQHQTTQQSYCPLAGGRLLDRDT
VTSIASYYVKTPAQVLLRWRRQNGNVVVARSYREARIISNFFVEDFELAAMDMDAAGGLGDGTRRREDP
HTYAG
```

```
>41284_seq_c1(32.6kDa)
MTSTTGEFPGIPSVSLNDGHSTPVLGLGVGEHSEAEAERFVAARLEAGYRHIDAAAVYGNEAAAQRAVS
ASGIPEEEIYVTKKLAVAQQGFGTSSDAARASLRRLGLYYVYLPHIHWPAGDAGMIIDSWGGHCCADQD
GVSRSIGVCCFEEHHSSTIIDLSFFTPAINQREHHPLLQQAEHNNTNYQYGIVTTYYGPLGVGVLLDHA
AVAGVAQAGGKTPAHVLLRWSIQLQNVVIAHSANNDRITSNLEVFDFEETDDMMAMHNLLGGGPRRRFD
METYTG
```

```
>42186_seq_c1(33.4kDa)
MTGFAGSPPAPAIYLNDEATMPVLGLGVFFLSDDETTRAVSAALFIGCRLIDTAYYYGNEFVVGRAAAA
SGVAREELFVTTKTYTADQGQTRSAEACHYSLDRLGLDLVDLYHNHWWAPPVGKYVDAMGNMIQQRGEE
ARRQIGSSSFTATNIENLRDLTFVVPAVNVRELYPLLNQDELAAANAQHVVVTASYSHLALQLLHDNPT
VTSIASEYVKTPAQVLRRWRLQLGRAVRVRSARPERENANFDVFDFFLAHHHMHAHGGLNDNTRVREDP
LTYAGT
```

```
>43465_seq_c2(39.4kDa)
MKALDVQIIGRVSRSGLTTWQGGSPYWFFGDYYATGAARDIRCRRRALVVTLADTYYIYGHGKSERQLE
EFLDHDRTETVVASKCYPVPPGDAVCKNRERAAAPRRQNQRRPRYNQHQPLPVVPDSVQMPQMRDDHDS
QDIGAAGVSRYSLARCRCCDAALGRPVQCRQVRSSLHHPDALLERVPFYELNNNTVIAYSPLHQQHLGG
KYSLNNLPGGVRALPPLFGTEELLRREPLLLTLATIAVDVDPPPAQVVLAWLINSPGVVDPAQASSVET
LEFNVAAADNELSAQDADALTDRARAFRPVSTGTHLTSHVREKVSRSDEPRQQRQNQQ
```

```
>45873_seq_c2(38.9kDa)
MYYLDVDIIGGVSSSGLGTWQFGSREWFGGDYYATGAARAIVKRARALTTTLFDTAYIYGLKKSEEQLG
EAFGDDRTETVVASKVAVVAPAPAVICKRERARARRRQLNNLPLYQNHQNHPVVPDSVQMPQMRMRLDS
QDIGAAGVSNYSLARMRAADAALGRPVQSRQVRSSLHHPDALEDLVMFFELENRIVIAYSPLHTQLLGG
GYGLENLPGGVRARNPLFGTENLRIIEPLLLTLAAIAVDVDAPPAQVVLWWLISLPGVVPIAGASSVSQ
LEFNVAAADNELSASSRDALTDAARAFRPVSTGLFLTSRYREKVSRRREEMHRHTYTPP
```

```
>45878_seq_c1(33.0kDa)
MTGFTGRQSQPSIILNDEMTVPTLGLGAAELSEDETERAVHAALIIGCRLIDTAAAYQNEAAVHRAIAA
SGQPTAMLFVDTKLATPDQGYTSSSDACAASLDRLGVDYVDRYHIHWWAPPVGVFVDAWGGIISRSGEG
HARRISVGNFTEEARISIIDLTFTAPAVNQIELHPLWNQDEHHKKKAQHNVTQQSYTPRPLGRLYDNST
RTRIAFEFTKTPAQVLLRLNLQLGLAVVARSAAAEHIHSNNDVDDFELALMMMDAAGQLDDQTRRRPDP
MTEAGS
```

```
>49250_seq_c2(39.8kDa)
MVHLANPERCGMVSCRCCGPSGLRLPASSSGHWAIFGFVNALSSQRDRLCKAGDLGITHFDLANNYGPP
PGSHQSEEFLLLRFDFAAYHDELITSTKAFFDMWPGPYGSGGGLSSLLLSLDSRLKLMGLFYVDIFPSH
HVDENTPTEETASDLAAAVQSGAALYVGSSSYSPERTQKVVVLLLEEKKPLLLLQPEYWLLNNWVKKGG
HLLTLQLNGVGGIADAPPAGGLLTGKYLNGIPFHSHMHRRGNKKRGLTPKMLTEAALNSLRLLNEMASQ
RGQDMAFMALSWLLKDDNVTSSLSGASRARQLEENVQHLINLTFSTSELAQIDQHIADAELLLNQASSD
K
```

The following five sequences could not be cloned due to cloning inefficiency:

```
>26539_seq_c1
MTTSTGHTSQPSIIRNDNMTMPTLGLFLAELEEDETERAVLAAHERGCRLIDTAAAYQNEAAVARAIAS
SGRPRHRLFTTTKLATPDQGFTKCQDADAASLSSLGVFYVDLYHIHWWAPPGGFQVDQWGGMIQSRGEG
HARSIGVICFTAEHLHHIIDLTFVTPAVCIIELHPLQNQDEYRKKKAQHNVIVTSYSPLPLGRLMDNDT
LTAIAAEFGKQPAQVLRLWRLHLGLAVVVRAHAAEAIHSSFDVDDFELAMHHMDAAAALDDRTRRRPDP
ETYAFS
```

```
>26535_seq_c2
MYYLDVDGIGGVSRIGLGTWQNGSMSWFYFDRGATGAARDCCCTAAAAGVALDDTAYNYGLGKSERQLG
HALGDHRTEVVVVSKKGPVPPEPYCIKRRERTSHSLHLLRRPPEYSIHQPRPVTPDSVQIPGMRDLLDS
GDIGGAGRSRHYLAWLAKADAALGRPVVSNQVASAHALPLDLELLVPFAFLENLIVIAPAALARLLLDG
GYGHVNRPGGAPALLPLAGTENLHIILPHLLTLRAIAADVDPYKAQVALAWLIWSPGVVDIPGVSVVTQ
LEENVHAADIELSALSTSADTDAPLAFRHVSTGRDLDDLVREKVSRR
```

```
>23919_seq_c1
MTESSGSSPAPSIILNDEDTVPVLGLGVDELSDDETETAVAAALEIGCNLIDTAAYYGNEAVVGRAIAA
SGVPRAHLFVTVKHATPDQGFTMSQEACHASLDRLGLDYVDLYLIHWWAPPVTKYVDAWGGMIQRRREG
HQRSISVTCFTAEHRYNNIDRTFVVPAVYIIELHPLLNQDELHDKIAQHTVTPQSYCPLALGFLHDNDT
VTHIASEYVKTPAQVLLLWNLGLGNAVVVRSALEELEAFQFDVFDGELAHHHMDAAGGLQDGTRLLLDP
HTYAG
```

```
>11719_seq_c1
MYTPTIINLQDGNVMPQLGPGWWQDSYEMVIIAINKALFVGQNSIDTAAYYYNEEGCGKALKNASVYRE
ELFTTTKLNNDDKNRPREALFHSLYKLQLDYLDRHHMHWPVPAADHAVAAPKGMQARQKGGLTRSIGVC
RFRIHHLQRLIDETGVTPVNNIEELLLLMQQHQLHACCDNHIIATFSPSPLAQGWFGVFFQGVNRDLAA
KYGPQPAIIVINNHLDHGLVVAPQSVTPSRDDENFMVWDFRDTKDELGFIAKLDGGGRLGPDPDQFGGG
```

```
>11553_seq_c1
MAIPAFGLGTFRLKDDVVISSVKTAHELGHNAIDTAQYYDNEAMVGAAAAEGGVPRHELYITTKWPIEN
LSKDKLIPSLWKSLQKLRTDVVVLTHIHWSSPPDEVSVEEFMQMLLEDKKFGLTREIQISNDTIPRWEA
AIAAVFADDDHTNQIEHSPYLQNRKVVDAAWQAGIHITSYWTLAYGKHLDDDVIDIIAADANATPAGVI
LWWAMGEGYSVIPSSTQREELYSNLSAQNLHLAAEDAKAIARLDCIDRLVSPEGLYPAWD
```

Bibliography

[1] "Uniprot: the universal protein knowledgebase in 2021," *Nucleic Acids Research*, vol. 49, no. D1, pp. D480–D489, 2021.

[2] L. Sanchez-Pulido and C. P. Ponting, "Extending the horizon of homology detection with coevolution-based structure prediction," *Journal of Molecular Biology*, p. 167106, 2021.

[3] S. Seemayer, M. Gruber, and J. Söding, "Ccmpred—fast and precise prediction of protein residue–residue contacts from correlated mutations," *Bioinformatics*, vol. 30, no. 21, pp. 3128–3130, 2014.

[4] N. Ketkar, "Introduction to keras," in *Deep learning with Python*, pp. 97–111, Springer, 2017.

[5] N. Ran, L. Zhao, Z. Chen, and J. Tao, "Recent applications of biocatalysis in developing green chemistry for chemical synthesis at the industrial scale," *Green Chem.*, vol. 10, no. 4, pp. 361–372, 2008.

[6] M. Margulies, M. Egholm, W. E. Altman, S. Attiya, J. S. Bader, L. A. Bemben, J. Berka, M. S. Braverman, Y.-J. Chen, Z. Chen, *et al.*, "Genome sequencing in microfabricated high-density picolitre reactors," *Nature*, vol. 437, no. 7057, pp. 376–380, 2005.

[7] E. L. Van Dijk, H. Auger, Y. Jaszczyszyn, and C. Thermes, "Ten years of next-generation sequencing technology," *Trends in genetics*, vol. 30, no. 9, pp. 418–426, 2014.

[8] N. H. G. R. Institute, "Dna sequencing costs: data from the nhgri genome sequencing program (gsp).," 2018.

[9] B. V. Jones, M. Begley, C. Hill, C. G. Gahan, and J. R. Marchesi, "Functional and comparative metagenomic analysis of bile salt hydrolase activity in the human gut microbiome," *Proceedings of the national academy of sciences*, vol. 105, no. 36, pp. 13580–13585, 2008.

[10] W. Xie, F. Wang, L. Guo, Z. Chen, S. M. Sievert, J. Meng, G. Huang, Y. Li, Q. Yan, S. Wu, *et al.*, "Comparative metagenomics of microbial communities inhabiting deep-sea hydrothermal vent chimneys with contrasting chemistries," *The ISME journal*, vol. 5, no. 3, pp. 414–426, 2011.

[11] R. Leinonen, R. Akhtar, E. Birney, L. Bower, A. Cerdeno-Tárraga, Y. Cheng, I. Cleland, N. Faruque, N. Goodgame, R. Gibson, *et al.*, "The european nucleotide archive," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D28–D31, 2010.

[12] P. W. Harrison, A. Ahamed, R. Aslam, B. T. Alako, J. Burgin, N. Buso, M. Courtot, J. Fan, D. Gupta, M. Haseeb, *et al.*, "The european nucleotide archive in 2020," *Nucleic acids research*, vol. 49, no. D1, pp. D82–D85, 2021.

[13] U. Consortium, "Activities at the universal protein resource (uniprot)," *Nucleic acids research*, vol. 42, no. D1, pp. D191–D198, 2014.

[14] D. D. Roumpeka, R. J. Wallace, F. Escalettes, I. Fotheringham, and M. Watson, "A review of bioinformatics tools for bio-prospecting from metagenomic sequence data," *Frontiers in genetics*, vol. 8, p. 23, 2017.

[15] A. Radzicka and R. Wolfenden, "A proficient enzyme," *Science*, vol. 267, no. 5194, pp. 90–93, 1995.

[16] C. A. Ouzounis and P. D. Karp, "Global properties of the metabolic map of escherichia coli," *Genome research*, vol. 10, no. 4, pp. 568–576, 2000.

[17] D. J. Pollard and J. M. Woodley, "Biocatalysis for pharmaceutical intermediates: the future is now," *TRENDS in Biotechnology*, vol. 25, no. 2, pp. 66–73, 2007.

[18] G. W. Huisman and S. J. Collier, "On the development of new biocatalytic processes for practical pharmaceutical synthesis," *Current opinion in chemical biology*, vol. 17, no. 2, pp. 284–292, 2013.

[19] N. Ran, L. Zhao, Z. Chen, and J. Tao, "Recent applications of biocatalysis in developing green chemistry for chemical synthesis at the industrial scale," *Green Chemistry*, vol. 10, no. 4, pp. 361–372, 2008.

[20] F. Hollmann, Y. Gumulya, C. Tolle, A. Liese, and O. Thum, "Evaluation of the laccase from myceliophthora thermophila as industrial biocatalyst for polymerization reactions," *Macromolecules*, vol. 41, no. 22, pp. 8520–8524, 2008.

[21] J. S. Wallace and C. J. Morrow, "Biocatalytic synthesis of polymers. synthesis of an optically active, epoxy-substituted polyester by lipase-catalyzed polymerization," *Journal of Polymer Science Part A: Polymer Chemistry*, vol. 27, no. 8, pp. 2553–2567, 1989.

[22] F. De Salas, I. Pardo, H. J. Salavagione, P. Aza, E. Amougi, J. Vind, A. T. Martinez, and S. Camarero, "Advanced synthesis of conductive polyaniline using laccase as biocatalyst," *PLoS One*, vol. 11, no. 10, p. e0164958, 2016.

[23] Y. Chung, Y. Ahn, M. Christwardana, H. Kim, and Y. Kwon, "Development of a glucose oxidase-based biocatalyst adopting both physical entrapment and crosslinking, and its use in biofuel cells," *Nanoscale*, vol. 8, no. 17, pp. 9201–9210, 2016.

[24] V. Kumar, N. Misra, N. K. Goel, R. Thakar, J. Gupta, and L. Varshney, "A horseradish peroxidase immobilized radiation grafted polymer matrix: a biocatalytic system for dye waste water treatment," *RSC advances*, vol. 6, no. 4, pp. 2974–2981, 2016.

[25] M. Memarpoor-Yazdi, H. R. Karbalaei-Heidari, and K. Khajeh, "Production of the renewable extremophile lipase: Valuable biocatalyst with potential usage in food industry," *Food and Bioproducts Processing*, vol. 102, pp. 153–166, 2017.

[26] B. S. Panwar and R. Trivedi, "Metagenomics: An era of throughput gene mining," in *Understanding Host-Microbiome Interactions-An Omics Approach*, pp. 41–54, Springer, 2017.

[27] K. Bastard, A. A. T. Smith, C. Vergne-Vaxelaire, A. Perret, A. Zaparucha, R. De Melo-Minardi, A. Mariage, M. Boutard, A. Debard, C. Lechaplais, *et al.*, "Revealing the hidden functional diversity of an enzyme family," *Nature chemical biology*, vol. 10, no. 1, pp. 42–49, 2014.

[28] Y. Li, S. Wang, R. Umarov, B. Xie, M. Fan, L. Li, and X. Gao, "DEEPre: sequence-based enzyme EC number prediction by deep learning," *Bioinformatics*, vol. 34, pp. 760–769, oct 2017.

[29] R. J. Roberts, "Identifying protein function—a call for community action," *PLoS biology*, vol. 2, no. 3, p. e42, 2004.

[30] L. Fernández-Arrojo, M.-E. Guazzaroni, N. López-Cortés, A. Beloqui, and M. Ferrer, "Metagenomic era for biocatalyst identification," *Current Opinion in Biotechnology*, vol. 21, no. 6, pp. 725–733, 2010.

[31] L.-L. Li, S. R. McCorkle, S. Monchy, S. Taghavi, and D. van der Lelie, "Bioprospecting metagenomes: glycosyl hydrolases for converting biomass," *Biotechnology for biofuels*, vol. 2, no. 1, pp. 1–11, 2009.

[32] S. Velikogne, V. Resch, C. Dertnig, J. H. Schrittwieser, and W. Kroutil, "Sequence-based in-silico discovery, characterisation, and biocatalytic application of a set of imine reductases," *ChemCatChem*, vol. 10, no. 15, p. 3236, 2018.

[33] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. Sonnhammer, *et al.*, "The pfam protein families database," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D138–D141, 2004.

[34] P. Vanacek, E. Sebestova, P. Babkova, S. Bidmanova, L. Daniel, P. Dvorak, V. Stepankova, R. Chaloupkova, J. Brezovsky, Z. Prokop, *et al.*, "Exploration of enzyme diversity by integrating bioinformatics with expression analysis and biochemical characterization," *Acs Catalysis*, vol. 8, no. 3, pp. 2402–2412, 2018.

[35] K. D. Pruitt, T. Tatusova, and D. R. Maglott, "Ncbi reference sequence (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins," *Nucleic acids research*, vol. 33, no. suppl_1, pp. D501–D504, 2005.

[36] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[37] J.-Y. Kim, J.-Y. Lee, Y.-S. Shin, and G.-J. Kim, "Mining and identification of a glucosidase family enzyme with high activity toward the plant extract indican," *Journal of Molecular Catalysis B: Enzymatic*, vol. 57, no. 1-4, pp. 284–291, 2009.

[38] D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, "Genbank," *Nucleic acids research*, vol. 41, no. D1, pp. D36–D42, 2012.

[39] D. Baud, J. W. Jeffries, T. S. Moody, J. M. Ward, and H. C. Hailes, "A metagenomics approach for new biocatalyst discovery: application to transaminases and the synthesis of allylic amines," *Green Chemistry*, vol. 19, no. 4, pp. 1134–1143, 2017.

[40] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped blast and psi-blast: a new generation of protein database search programs," *Nucleic acids research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[41] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[42] B. Rost, "Twilight zone of protein sequence alignments," *Protein Engineering, Design and Selection*, vol. 12, pp. 85–94, feb 1999.

[43] B. Rost, "Enzyme function less conserved than anticipated," *Journal of Molecular Biology*, vol. 318, pp. 595–608, apr 2002.

[44] W. Tian and J. Skolnick, "How well is enzyme function conserved as a function of pairwise sequence identity?," *Journal of molecular biology*, vol. 333, no. 4, pp. 863–882, 2003.

[45] J. M. Jez, T. G. Flynn, and T. M. Penning, "A new nomenclature for the aldo-keto reductase superfamily," *Biochemical pharmacology*, vol. 54, no. 6, pp. 639–647, 1997.

[46] A. Babtie, N. Tokuriki, and F. Hollfelder, "What makes an enzyme promiscuous?," *Current opinion in chemical biology*, vol. 14, no. 2, pp. 200–207, 2010.

[47] S. D. Brown and P. C. Babbitt, "New insights about enzyme evolution from large scale studies of sequence and structure relationships," *Journal of Biological Chemistry*, vol. 289, pp. 30221–30228, sep 2014.

[48] F. Baier, J. N. Copp, and N. Tokuriki, "Evolution of enzyme superfamilies: Comprehensive exploration of sequence–function relationships," *Biochemistry*, vol. 55, pp. 6375–6388, nov 2016.

[49] L. De Ferrari, S. Aitken, J. van Hemert, and I. Goryanin, "Multi-label prediction of enzyme classes using interpro signatures," *Machine Learning in Systems Biology*, p. 123, 2010.

[50] N. M. Luscombe, D. Greenbaum, and M. Gerstein, "What is bioinformatics? a proposed definition and overview of the field," *Methods of information in medicine*, vol. 40, no. 04, pp. 346–358, 2001.

[51] J. Mistry, S. Chuguransky, L. Williams, M. Qureshi, G. A. Salazar, E. L. Sonnhammer, S. C. Tosatto, L. Paladin, S. Raj, L. J. Richardson, *et al.*, “Pfam: The protein families database in 2021,” *Nucleic Acids Research*, vol. 49, no. D1, pp. D412–D419, 2021.

[52] S. Hunter, R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bork, U. Das, L. Daugherty, L. Duquenne, *et al.*, “Interpro: the integrative protein signature database,” *Nucleic acids research*, vol. 37, no. suppl_1, pp. D211–D215, 2009.

[53] A. Marchler-Bauer, S. Lu, J. B. Anderson, F. Chitsaz, M. K. Derbyshire, C. DeWeese-Scott, J. H. Fong, L. Y. Geer, R. C. Geer, N. R. Gonzales, *et al.*, “Cdd: a conserved domain database for the functional annotation of proteins,” *Nucleic acids research*, vol. 39, no. suppl_1, pp. D225–D229, 2010.

[54] E. M. Zdobnov and R. Apweiler, “Interproscan–an integration platform for the signature-recognition methods in interpro,” *Bioinformatics*, vol. 17, no. 9, pp. 847–848, 2001.

[55] A. Bairoch, “The enzyme database in 2000,” *Nucleic acids research*, vol. 28, no. 1, pp. 304–305, 2000.

[56] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The protein data bank,” *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

[57] J. Pérez, M. Arenas, and C. Gutierrez, “Semantics and complexity of sparql,” *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 3, pp. 1–45, 2009.

[58] J. Kim, D. Kyung, H. Yun, B.-K. Cho, J.-H. Seo, M. Cha, and B.-G. Kim, “Cloning and characterization of a novel β-transaminase from mesorhizobium sp. strain luk: a new biocatalyst for the synthesis of enantiomerically pure β-amino acids,” *Applied and environmental microbiology*, vol. 73, no. 6, pp. 1772–1782, 2007.

[59] K. Wu, K. Zheng, L. Xiong, Z. Yang, Z. Jiang, X. Meng, and L. Shao, “Efficient synthesis of an antiviral drug intermediate using an enhanced short-chain dehydrogenase in an aqueous-organic solvent system,” *Applied microbiology and biotechnology*, vol. 103, no. 11, pp. 4417–4427, 2019.

[60] R. B. Hamed, L. Henry, J. R. Gomez-Castellanos, J. Mecinovic, C. Ducho, J. L. Sorensen, T. D. Claridge, and C. J. Schofield, “Crotonase catalysis enables flexible production of functionalized prolines and carbapenams,” *Journal of the American Chemical Society*, vol. 134, no. 1, pp. 471–479, 2012.

[61] K. Honarmand Ebrahimi, J. S. Rowbotham, J. McCullagh, and W. S. James, “Mechanism of diol dehydration by a promiscuous radical-sam enzyme homologue of the antiviral enzyme viperin (rsad2),” *ChemBioChem*, vol. 21, no. 11, pp. 1605–1612, 2020.

[62] V. Höllrigl, F. Hollmann, A. C. Kleeb, K. Buehler, and A. Schmid, “Tadh, the thermostable alcohol dehydrogenase from thermus sp. atn1: a versatile new biocatalyst for organic synthesis,” *Applied microbiology and biotechnology*, vol. 81, no. 2, pp. 263–273, 2008.

[63] T. Kim, R. Flick, J. Brunzelle, A. Singer, E. Evdokimova, G. Brown, J. C. Joo, G. A. Minasov, W. F. Anderson, R. Mahadevan, *et al.*, “Novel aldo-keto reductases for the biocatalytic conversion of 3-hydroxybutanal to 1, 3-butanediol: structural and biochemical studies,” *Applied and environmental microbiology*, vol. 83, no. 7, pp. e03172–16, 2017.

[64] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden, “Blast+: architecture and applications,” *BMC bioinformatics*, vol. 10, no. 1, pp. 1–9, 2009.

[65] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[66] P. Rice, I. Longden, and A. Bleasby, “Emboss: the european molecular biology open software suite,” *Trends in genetics*, vol. 16, no. 6, pp. 276–277, 2000.

[67] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, *et al.*, “Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega,” *Molecular systems biology*, vol. 7, no. 1, p. 539, 2011.

[68] D. G. Higgins, J. D. Thompson, and T. J. Gibson, “[22] using clustal for multiple sequence alignments,” *Methods in enzymology*, vol. 266, pp. 383–402, 1996.

[69] F. Sievers and D. G. Higgins, “Clustal omega,” *Current protocols in bioinformatics*, vol. 48, no. 1, pp. 3–13, 2014.

[70] R. D. Finn, J. Clements, and S. R. Eddy, “Hmmer web server: interactive sequence similarity searching,” *Nucleic acids research*, vol. 39, no. suppl_2, pp. W29–W37, 2011.

[71] I. Lobo, “Basic local alignment search tool (blast),” *Nat Educ*, vol. 1, no. 1, 2008.

[72] J. Sullivan and P. Joyce, “Model selection in phylogenetics,” *Annu. Rev. Ecol. Evol. Syst.*, vol. 36, pp. 445–466, 2005.

[73] S. K. Hanks, A. M. Quinn, and T. Hunter, “The protein kinase family: conserved features and deduced phylogeny of the catalytic domains,” *Science*, vol. 241, no. 4861, pp. 42–52, 1988.

[74] H. J. Atkinson, J. H. Morris, T. E. Ferrin, and P. C. Babbitt, “Using sequence similarity networks for visualization of relationships across diverse protein superfamilies,” *PLoS ONE*, vol. 4, p. e4345, feb 2009.

[75] M. Sadowski and D. Jones, “The sequence–structure relationship and protein function prediction,” *Current opinion in structural biology*, vol. 19, no. 3, pp. 357–362, 2009.

[76] M. P. Kötzler, S. M. Hancock, and S. G. Withers, “Glycosidases: Functions, families and folds,” *eLS*, 2014.

[77] D. Devos and A. Valencia, “Practical limits of function prediction,” *Proteins: Structure, Function, and Bioinformatics*, vol. 41, no. 1, pp. 98–107, 2000.

[78] K. Arnold, L. Bordoli, J. Kopp, and T. Schwede, “The swiss-model workspace: a web-based environment for protein structure homology modelling,” *Bioinformatics*, vol. 22, no. 2, pp. 195–201, 2006.

[79] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moult, “Critical assessment of methods of protein structure prediction (casp)—round xiii,” *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1011–1020, 2019.

[80] Y. Zhang and J. Skolnick, “Tm-align: a protein structure alignment algorithm based on the tm-score,” *Nucleic acids research*, vol. 33, no. 7, pp. 2302–2309, 2005.

[81] J. Xu and Y. Zhang, “How significant is a protein structure similarity with tm-score= 0.5?,” *Bioinformatics*, vol. 26, no. 7, pp. 889–895, 2010.

[82] Y. Zhang, “I-tasser server for protein 3d structure prediction,” *BMC bioinformatics*, vol. 9, no. 1, pp. 1–8, 2008.

[83] I. El Naqa and M. J. Murphy, “What is machine learning?,” in *machine learning in radiation oncology*, pp. 3–11, Springer, 2015.

[84] S. H. Zanakis and J. R. Evans, “Heuristic “optimization”: Why, when, and how to use it,” *Interfaces*, vol. 11, no. 5, pp. 84–91, 1981.

[85] E. Kapun and F. Tsarev, “De bruijn superwalk with multiplicities problem is np-hard,” in *BMC bioinformatics*, vol. 14, pp. 1–4, Springer, 2013.

[86] D. T. Jones, “Protein secondary structure prediction based on position-specific scoring matrices,” *Journal of molecular biology*, vol. 292, no. 2, pp. 195–202, 1999.

[87] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, “Highly accurate protein structure prediction with alphafold,” *Nature*, p. 1, 2021.

[88] Y. Li, S. Wang, R. Umarov, B. Xie, M. Fan, L. Li, and X. Gao, “Deepre: sequence-based enzyme ec number prediction by deep learning,” *Bioinformatics*, vol. 34, no. 5, pp. 760–769, 2018.

[89] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, no. 1, pp. 43–62, 1997.

[90] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[91] S. Sharma and S. Sharma, "Activation functions in neural networks," *Towards Data Science*, vol. 6, no. 12, pp. 310–316, 2017.

[92] S. Mannor, D. Peleg, and R. Rubinstein, "The cross entropy method for classification," in *Proceedings of the 22nd international conference on Machine learning*, pp. 561–568, 2005.

[93] Y. Xia and J. Wang, "A one-layer recurrent neural network for support vector machine learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 2, pp. 1261–1269, 2004.

[94] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pp. 1–2, IEEE, 2018.

[95] L. Bottou, "Stochastic gradient descent tricks," in *Neural networks: Tricks of the trade*, pp. 421–436, Springer, 2012.

[96] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[97] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[98] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 490–497, 2014.

[99] F. Glover, "Tabu search: A tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.

[100] F. Glover and M. Laguna, "Tabu search," in *Handbook of combinatorial optimization*, pp. 2093–2229, Springer, 1998.

[101] M. Malek, M. Guruswamy, M. Pandya, and H. Owens, "Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem," *Annals of Operations Research*, vol. 21, no. 1, pp. 59–84, 1989.

[102] A. Hertz and D. de Werra, "Using tabu search techniques for graph coloring," *Computing*, vol. 39, no. 4, pp. 345–351, 1987.

[103] Y. Wang, J.-K. Hao, F. Glover, and Z. Lü, “A tabu search based memetic algorithm for the maximum diversity problem,” *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 103–114, jan 2014.

[104] L. Serrano, “Synthetic biology: promises and challenges,” 2007.

[105] S. C. Wheelwright and K. B. Clark, “Accelerating the design-build-test cycle for effective product development,” *International Marketing Review*, 1994.

[106] W. Gao, A. Rzewski, H. Sun, P. D. Robbins, and A. Gambotto, “Upgene: application of a web-based dna codon optimization algorithm,” *Biotechnology progress*, vol. 20, no. 2, pp. 443–448, 2004.

[107] S. R. Kotra, J. Peravali, S. Yanamadala, A. Kumar, K. Samba Siva Rao, and K. Pulicherla, “Large scale production of soluble recombinant staphylokinase variant from cold shock expression system using iptg inducible e. coli bl21 (de3),” *Int J Bio-Sci Bio-Technol*, vol. 5, pp. 107–116, 2013.

[108] P. Carbonell, A. J. Jervis, C. J. Robinson, C. Yan, M. Dunstan, N. Swainston, M. Vinaixa, K. A. Hollywood, A. Currin, N. J. Rattray, *et al.*, “An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals,” *Communications biology*, vol. 1, no. 1, pp. 1–10, 2018.

[109] J. A. McLaughlin, C. J. Myers, Z. Zundel, G. Misirli, M. Zhang, I. D. Ofiteru, A. Goñi Moreno, and A. Wipat, “Synbiohub: A standards-enabled design repository for synthetic biology,” *ACS synthetic biology*, 2018.

[110] C. Vilanova and M. Porcar, “igem 2.0—refoundations for engineering biology,” *Nature biotechnology*, vol. 32, no. 5, pp. 420–424, 2014.

[111] T. Knight, “Idempotent vector design for standard assembly of biobricks,” 2003.

[112] M. Galdzicki, K. P. Clancy, E. Oberortner, M. Pocock, J. Y. Quinn, C. A. Rodriguez, N. Roehner, M. L. Wilson, L. Adam, J. C. Anderson, *et al.*, “The synthetic biology open language (sbol) provides a community standard for communicating designs in synthetic biology,” *Nature biotechnology*, vol. 32, no. 6, pp. 545–550, 2014.

[113] W. R. Pearson, “[5] rapid and sensitive sequence comparison with fastp and fasta,” 1990.

[114] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, “Genbank,” *Nucleic acids research*, vol. 39, no. suppl_1, pp. D32–D37, 2010.

[115] H. Baig, P. Fontanarrosa, V. Kulkarni, J. A. McLaughlin, P. Vaidyanathan, B. Bartley, J. Beal, M. Crowther, T. E. Gorochowski, R. Grünberg, *et al.*, “Synthetic biology open language (sbol) version 3.0. 0,” *Journal of integrative bioinformatics*, vol. 17, no. 2-3, 2020.

[116] C. Madsen, A. G. Moreno, P. Umesh, Z. Palchick, N. Roehner, C. Atallah, B. Bartley, K. Choi, R. S. Cox, T. Gorochowski, *et al.*, "Synthetic biology open language (sbol) version 2.3," *Journal of integrative bioinformatics*, vol. 16, no. 2, 2019.

[117] "Sep19." `https://github.com/SynBioDex/SEPs/blob/master/sep_019.md`.

[118] "Sep21." `https://github.com/SynBioDex/SEPs/blob/master/sep_021.md`.

[119] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao, "Prov-o: The prov ontology," 2013.

[120] M. Zhang, J. A. McLaughlin, A. Wipat, and C. J. Myers, "Sboldesigner 2: an intuitive tool for structural genetic design," *ACS synthetic biology*, vol. 6, no. 7, pp. 1150–1160, 2017.

[121] A. A. Nielsen, B. S. Der, J. Shin, P. Vaidyanathan, V. Paralanov, E. A. Strychalski, D. Ross, D. Densmore, and C. A. Voigt, "Genetic circuit design automation," *Science*, vol. 352, no. 6281, 2016.

[122] C. J. Myers, N. Barker, K. Jones, H. Kuwahara, C. Madsen, and N.-P. D. Nguyen, "ibiosim: a tool for the analysis and design of genetic circuits," *Bioinformatics*, vol. 25, no. 21, pp. 2848–2849, 2009.

[123] M. Crowther, L. Grozinger, M. Pocock, C. P. Taylor, J. A. McLaughlin, G. Mısırlı, B. A. Bartley, J. Beal, A. Goñi-Moreno, and A. Wipat, "Shortbol: a language for scripting designs for engineered biological systems using synthetic biology open language (sbol)," *ACS synthetic biology*, vol. 9, no. 4, pp. 962–966, 2020.

[124] J. A. McLaughlin, C. J. Myers, Z. Zundel, N. Wilkinson, C. Atallah, and A. Wipat, "sboljs: Bringing the synthetic biology open language to the web browser," *ACS synthetic biology*, vol. 8, no. 1, pp. 191–193, 2018.

[125] G. Yanez Feliu, B. Earle Gomez, V. Codoceo Berrocal, M. Munoz Silva, I. N. Nunez, T. F. Matute, A. Arce Medina, G. Vidal, C. Vidal Cespedes, J. Dahlin, *et al.*, "Flapjack: Data management and analysis for genetic circuit characterization," *ACS Synthetic Biology*, vol. 10, no. 1, pp. 183–191, 2020.

[126] F. Baier, J. Copp, and N. Tokuriki, "Evolution of enzyme superfamilies: comprehensive exploration of sequence–function relationships," *Biochemistry*, vol. 55, no. 46, pp. 6375–6388, 2016.

[127] W. R. Pearson and M. L. Sierk, "The limits of protein sequence comparison?," *Current opinion in structural biology*, vol. 15, no. 3, pp. 254–260, 2005.

[128] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, *et al.*, "Uniprot: the universal protein knowledgebase," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D115–D119, 2004.

[129] Uniprot Consortium, "Uniprotkb/swiss-prot uniprot release 2019/02." https://www.uniprot.org/statistics/Swiss-Prot%202019_02, Feb. 2019.

[130] Uniprot Consortium, "Uniprotkb/trembl uniprot release 2019/02." https://www.uniprot.org/statistics/TrEMBL%202019_02, Feb. 2019.

[131] T. Lima, A. H. Auchincloss, E. Coudert, G. Keller, K. Michoud, C. Rivoire, V. Bulliard, E. de Castro, C. Lachaize, D. Baratin, I. Phan, L. Bougueleret, and A. Bairoch, "HAMAP: a database of completely sequenced microbial proteome sets and manually curated microbial protein families in UniProtKB/swiss-prot," *Nucleic Acids Research*, vol. 37, pp. D471–D478, jan 2009.

[132] A. Gattiker, K. Michoud, C. Rivoire, A. H. Auchincloss, E. Coudert, T. Lima, P. Kersey, M. Pagni, C. J. Sigrist, C. Lachaize, A.-L. Veuthey, E. Gasteiger, and A. Bairoch, "Automated annotation of microbial proteomes in SWISS-PROT," *Computational Biology and Chemistry*, vol. 27, pp. 49–58, feb 2003.

[133] A. M. Schnoes, S. D. Brown, I. Dodevski, and P. C. Babbitt, "Annotation error in public databases: misannotation of molecular function in enzyme superfamilies," *PLoS Comput Biol*, vol. 5, no. 12, p. e1000605, 2009.

[134] H. Bagheri, A. J. Severin, and H. Rajan, "Detecting and correcting misclassified sequences in the large-scale public databases," *Bioinformatics*, vol. 36, no. 18, pp. 4699–4705, 2020.

[135] S. Whelan, "New approaches to phylogenetic tree search and their application to large numbers of protein alignments," *Systematic biology*, vol. 56, no. 5, pp. 727–740, 2007.

[136] J. A. Gerlt, J. T. Bouvier, D. B. Davidson, H. J. Imker, B. Sadkhin, D. R. Slater, and K. L. Whalen, "Enzyme function initiative-enzyme similarity tool (efi-est): A web tool for generating protein sequence similarity networks," *Biochimica Et Biophysica Acta (BBA)-Proteins and Proteomics*, vol. 1854, no. 8, pp. 1019–1037, 2015.

[137] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, "Direct-coupling analysis of residue coevolution captures native contacts across many protein families," *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. E1293–E1301, 2011.

[138] J. Moult, "A decade of casp: progress, bottlenecks and prognosis in protein structure prediction," *Current opinion in structural biology*, vol. 15, no. 3, pp. 285–289, 2005.

[139] J. Schaarschmidt, B. Monastyrskyy, A. Kryshtafovych, and A. M. Bonvin, "Assessment of contact predictions in casp12: Co-evolution and deep learning coming of age," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, pp. 51–66, 2018.

[140] F. Pazos, M. Helmer-Citterich, G. Ausiello, and A. Valencia, "Correlated mutations contain information about protein-protein interaction," *Journal of molecular biology*, vol. 271, no. 4, pp. 511–523, 1997.

[141] V. H. Salinas and R. Ranganathan, "Coevolution-based inference of amino acid interactions underlying protein function," *eLife*, vol. 7, jul 2018.

[142] B.-C. Lee, K. Park, and D. Kim, "Analysis of the residue–residue coevolution network and the functionally important residues in proteins," *Proteins: Structure, Function, and Bioinformatics*, vol. 72, no. 3, pp. 863–872, 2008.

[143] R. S. Dwyer, D. P. Ricci, L. J. Colwell, T. J. Silhavy, and N. S. Wingreen, "Predicting functionally informative mutations in escherichia coli bama using evolutionary covariance analysis," *Genetics*, pp. genetics–113, 2013.

[144] Y. Lee, J. Mick, C. Furdui, and L. J. Beamer, "A coevolutionary residue network at the site of a functionally important conformational change in a phosphohexomutase enzyme family," *PLoS One*, vol. 7, no. 6, p. e38114, 2012.

[145] T. A. Hopf and D. S. Marks, "Protein structures, interactions and function from evolutionary couplings," in *From Protein Structure to Function with Bioinformatics*, pp. 37–58, Springer, 2017.

[146] S. Wang, S. Sun, Z. Li, R. Zhang, and J. Xu, "Accurate de novo prediction of protein contact map by ultra-deep learning model," *PLoS computational biology*, vol. 13, no. 1, p. e1005324, 2017.

[147] S. Seemayer, M. Gruber, and J. Söding, "Ccmpred wiki https://github.com/soedinglab/ccmpred/wiki/faq," 2014.

[148] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[149] D. de Juan, F. Pazos, and A. Valencia, "Emerging methods in protein coevolution," *Nature Reviews Genetics*, vol. 14, pp. 249–261, mar 2013.

[150] B. Schwikowski, P. Uetz, and S. Fields, "A network of protein–protein interactions in yeast," *Nature biotechnology*, vol. 18, no. 12, p. 1257, 2000.

[151] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome research*, vol. 13, no. 11, pp. 2498–2504, 2003.

[152] C. Giustini, M. Graindorge, D. Cobessi, S. Crouzy, A. Robin, G. Curien, and M. Matringe, "Tyrosine metabolism: identification of a key residue in the acquisition of prephenate aminotransferase activity by 1β aspartate aminotransferase," *The FEBS journal*, vol. 286, no. 11, pp. 2118–2134, 2019.

[153] A. M. Grishin, E. Ajamian, L. Zhang, I. Rouiller, M. Bostina, and M. Cygler, "Protein-protein interactions in the beta-oxidation part of the phenylacetate utilization pathway crystal structure of the paaf-paag hydratase-isomerase complex," *Journal of Biological Chemistry*, vol. 287, no. 45, pp. 37986–37996, 2012.

[154] D. Tan, W. M. Crabb, W. B. Whitman, and L. Tong, "Crystal structure of dmdd, a crotonase superfamily enzyme that catalyzes the hydration and hydrolysis of methylthioacryloyl-coa," *PloS one*, vol. 8, no. 5, p. e63870, 2013.

[155] M. N. Price, P. S. Dehal, and A. P. Arkin, "Fasttree 2–approximately maximum-likelihood trees for large alignments," *PloS one*, vol. 5, no. 3, p. e9490, 2010.

[156] T. G. Vaughan, "Icytree: rapid browser-based visualization for phylogenetic trees and networks," *Bioinformatics*, vol. 33, no. 15, pp. 2392–2394, 2017.

[157] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Generalized louvain method for community detection in large networks," in *2011 11th international conference on intelligent systems design and applications*, pp. 88–93, IEEE, 2011.

[158] F. L. Simonetti, E. Teppa, A. Chernomoretz, M. Nielsen, and C. Marino Buslje, "Mistic: mutual information server to infer coevolution," *Nucleic acids research*, vol. 41, no. W1, pp. W8–W14, 2013.

[159] M. Drosou and E. Pitoura, "Diverse set selection over dynamic data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1102–1116, 2013.

[160] R. Martí, M. Gallego, A. Duarte, and E. G. Pardo, "Heuristics and metaheuristics for the maximum diversity problem," *Journal of Heuristics*, vol. 19, no. 4, pp. 591–615, 2013.

[161] C.-C. Kuo, F. Glover, and K. S. Dhir, "Analyzing and modeling the maximum diversity problem by zero-one programming," *Decision Sciences*, vol. 24, no. 6, pp. 1171–1185, 1993.

[162] R. K. Kincaid, "Good solutions to discrete noxious location problems via metaheuristics," *Annals of Operations Research*, vol. 40, no. 1, pp. 265–281, 1992.

[163] K. Katayama and H. Narihisa, "An evolutionary approach for the maximum diversity problem," in *Recent advances in memetic algorithms*, pp. 31–47, Springer, 2005.

[164] M. Drosou and E. Pitoura, "Diverse set selection over dynamic data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 1102–1116, may 2014.

[165] A. Duarte and R. Martí, "Tabu search and grasp for the maximum diversity problem," *European Journal of Operational Research*, vol. 178, no. 1, pp. 71–84, 2007.

[166] E. Erkut, Y. Ülküsal, and O. Yeniçerioğlu, "A comparison of p-dispersion heuristics," *Computers & operations research*, vol. 21, no. 10, pp. 1103–1113, 1994.

[167] L. D. Ferrari, S. Aitken, J. van Hemert, and I. Goryanin, "EnzML: multi-label prediction of enzyme classes using InterPro signatures," *BMC Bioinformatics*, vol. 13, no. 1, p. 61, 2012.

[168] C. Caso and M. Angeles gil, "The gini-simpson index of diversity: estimation in the stratified sampling," *Communications in Statistics-Theory and Methods*, vol. 17, no. 9, pp. 2981–2995, 1988.

[169] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[170] R. C. Guiasu and S. Guiasu, "The weighted gini-simpson index: revitalizing an old index of biodiversity," *International Journal of Ecology*, vol. 2012, 2012.

[171] J. D. Bloom, M. M. Meyer, P. Meinhold, C. R. Otey, D. MacMillan, and F. H. Arnold, "Evolving strategies for enzyme engineering," *Current opinion in structural biology*, vol. 15, no. 4, pp. 447–452, 2005.

[172] A. Dubey and A. Verma, "Enzyme engineering for enzyme activity improvement," in *Enzymes in Food Biotechnology*, pp. 675–689, Elsevier, 2019.

[173] D. A. Estell, T. P. Graycar, and J. A. Wells, "Engineering an enzyme by site-directed mutagenesis to be resistant to chemical oxidation.," *Journal of Biological Chemistry*, vol. 260, no. 11, pp. 6518–6521, 1985.

[174] M. T. Reetz, P. Soni, L. Fernandez, Y. Gumulya, and J. D. Carballeira, "Increasing the stability of an enzyme toward hostile organic solvents by directed evolution based on iterative saturation mutagenesis using the b-fit method," *Chemical Communications*, vol. 46, no. 45, pp. 8657–8658, 2010.

[175] K. Chen and F. H. Arnold, "Enzyme engineering for nonaqueous solvents: random mutagenesis to enhance activity of subtilisin e in polar organic media," *Bio/Technology*, vol. 9, no. 11, pp. 1073–1077, 1991.

[176] Z. Costello and H. G. Martin, "How to hallucinate functional proteins,"

[177] C. Wan and D. T. Jones, "Protein function prediction is improved by creating synthetic feature samples with generative adversarial networks," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 540–550, 2020.

[178] D. Repecka, V. Jauniskis, L. Karpus, E. Rembeza, I. Rokaitis, J. Zrimec, S. Poviloniene, A. Laurynenas, S. Viknander, W. Abuajwa, *et al.*, "Expanding functional protein sequence spaces using generative adversarial networks," *Nature Machine Intelligence*, vol. 3, no. 4, pp. 324–333, 2021.

[179] X. Ding, Z. Zou, and C. L. Brooks III, "Deciphering protein evolution and fitness landscapes with latent space models," *Nature communications*, vol. 10, no. 1, pp. 1–13, 2019.

[180] L. J. McGuffin, K. Bryson, and D. T. Jones, "The psipred protein structure prediction server," *Bioinformatics*, vol. 16, no. 4, pp. 404–405, 2000.

[181] T. Kosciolek and D. T. Jones, “Accurate contact predictions using covariation techniques and machine learning,” *Proteins: Structure, Function, and Bioinformatics*, vol. 84, pp. 145–151, 2016.

[182] A. Gutteridge, G. J. Bartlett, and J. M. Thornton, “Using a neural network and spatial clustering to predict the location of active sites in enzymes,” *Journal of molecular biology*, vol. 330, no. 4, pp. 719–734, 2003.

[183] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” *arXiv preprint arXiv:1511.05644*, 2015.

[184] Z. Costello and H. G. Martin, “How to hallucinate functional proteins,” *arXiv preprint arXiv:1903.00458*, 2019.

[185] N. Redaschi, U. Consortium, *et al.*, “Uniprot in rdf: tackling data integration and distributed annotation with the semantic web,” *Nature precedings*, pp. 1–1, 2009.

[186] H. Luo and H. Nijveen, “Understanding and identifying amino acid repeats,” *Briefings in bioinformatics*, vol. 15, no. 4, pp. 582–591, 2014.

[187] J. Yang, R. Yan, A. Roy, D. Xu, J. Poisson, and Y. Zhang, “The i-tasser suite: protein structure and function prediction,” *Nature methods*, vol. 12, no. 1, pp. 7–8, 2015.

[188] A. Roy, A. Kucukural, and Y. Zhang, “I-tasser: a unified platform for automated protein structure and function prediction,” *Nature protocols*, vol. 5, no. 4, pp. 725–738, 2010.

[189] J. Yang, Y. Wang, and Y. Zhang, “Resq: an approach to unified estimation of b-factor and residue-specific error in protein structure prediction,” *Journal of molecular biology*, vol. 428, no. 4, pp. 693–701, 2016.

[190] F. H.-F. Leung, H.-K. Lam, S.-H. Ling, and P. K.-S. Tam, “Tuning of the structure and parameters of a neural network using an improved genetic algorithm,” *IEEE Transactions on Neural networks*, vol. 14, no. 1, pp. 79–88, 2003.

[191] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu, “Uniref: comprehensive and non-redundant uniprot reference clusters,” *Bioinformatics*, vol. 23, no. 10, pp. 1282–1288, 2007.

[192] W. Li and A. Godzik, “Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences,” *Bioinformatics*, vol. 22, no. 13, pp. 1658–1659, 2006.

[193] L. S. Johnson, S. R. Eddy, and E. Portugaly, “Hidden markov model speed heuristic and iterative hmm search procedure,” *BMC bioinformatics*, vol. 11, no. 1, p. 431, 2010.

[194] M. Grinberg, *Flask web development: developing web applications with python.* " O’Reilly Media, Inc.", 2018.

[195] C. Boettiger, "An introduction to docker for reproducible research," *ACM SIGOPS Operating Systems Review*, vol. 49, no. 1, pp. 71–79, 2015.

[196] J. Webber, "A programmatic introduction to neo4j," in *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity*, pp. 217–218, 2012.

[197] K. Degtyarenko, P. De Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcántara, M. Darsow, M. Guedj, and M. Ashburner, "Chebi: a database and ontology for chemical entities of biological interest," *Nucleic acids research*, vol. 36, no. suppl_1, pp. D344–D350, 2007.

[198] U. Urquiza-García, T. Zieliński, and A. J. Millar, "Better research by efficient sharing: evaluation of free management platforms for synthetic biology designs," *Synthetic Biology*, vol. 4, no. 1, p. ysz016, 2019.

[199] R. Vine, "Google scholar," *Journal of the Medical Library Association*, vol. 94, no. 1, p. 97, 2006.

[200] G. L. Holliday, S. D. Brown, D. Mischel, B. J. Polacco, and P. C. Babbitt, "A strategy for large-scale comparison of evolutionary-and reaction-based classifications of enzyme function," *Database*, vol. 2020, 2020.

[201] E. Rembeza and M. K. Engqvist, "Experimental and computational investigation of enzyme functional annotations uncovers misannotation in the ec 1.1. 3.15 enzyme class," *PLoS computational biology*, vol. 17, no. 9, p. e1009446, 2021.

[202] W. P. Russ, M. Figliuzzi, C. Stocker, P. Barrat-Charlaix, M. Socolich, P. Kast, D. Hilvert, R. Monasson, S. Cocco, M. Weigt, *et al.*, "An evolution-based model for designing chorismate mutase enzymes," *Science*, vol. 369, no. 6502, pp. 440–445, 2020.

[203] R. Furukawa, W. Toma, K. Yamazaki, and S. Akanuma, "Ancestral sequence reconstruction produces thermally stable enzymes with mesophilic enzyme-like catalytic properties," *Scientific reports*, vol. 10, no. 1, pp. 1–13, 2020.