Fall 2023

Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-ADDTI) Modeller10.4 Integrating Structure-Function Prediction Modules

Grant Galileo Jacobson

Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI) Modeller10.4 Integrating Structure-Function Prediction Modules

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Grant Galileo Jacobson

December 2023

The Designated Project Committee Approves the Project Titled

Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI) Modeller10.4 Integrating Structure-Function Prediction Modules

by

Grant Galileo Jacobson

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

December 2023

Dr. Leonard Wesley	Department of Computer Science
Dr. William Andreopoulos	Department of Computer Science
Dr. Cleber Ouverney	Department of Biology

Abstract

Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI) Modeller10.4 Integrating Structure-Function Prediction Modules

By Grant Galileo Jacobson

In the population of adult human patients who over express Beta and Tau Amyloids, it is unclear why 40% of them do not have Alzheimer's Disease (AD), when all patients with AD have an overexpression of Beta and Tau Amyloids. The MCI-AD-DTI project's epigenetic pipeline is an evolving computation tool that seeks epigenetic-related information related to the observed disparity. The MCI-AD-DTI's epigenetic pipeline's ability to identify mutations currently relies solely on PyPDB for verification of its protein functionality evaluation. The assessment process of the industry standard application, Modeller10.4, is independent from the current epigenetic pipeline's protein evaluation algorithm. Thus, this project concludes that integrating Modeller10.4, with its protein modeling, would contribute through an increase in the ability to distinguish between which Single Nucleotide Polymorphic (SNP) mutations change functionality.

Keywords: Alzheimer's Disease, Mild Cognitive Impairment, Epigenetics, Modeller, SNP

ACKNOWLEDGEMENT

MODELLER (copyright © 1989-2023 Andrej Sali) is maintained by Ben Webb at the Departments of Biopharmaceutical Sciences and Pharmaceutical Chemistry, and California Institute for Quantitative Biomedical Research, Mission Bay Byers Hall, University of California San Francisco, San Francisco, CA 94143, USA.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Alzheimer's Disease (AD) is responsible for 70% of the reported cases of dementia worldwide [1]. AD has been diagnosed by the two proteins amyloid beta (Aβ) and tau for many years [2]. Aβ are peptides which come from the amyloid precursor protein that promote the growth of neurons [2]. In AD, amyloid beta is heavily clustered in the brain. This clustering disrupts Aβs true function and leads to neural degeneration. Tau supports microtubules which help neurons stay healthy. In a brain that has AD, tau gets tangled up in the part of the brain that is associated with memory. After numerous studies, it has been determined that the onset of AD cannot be solely caused by high levels of Aβ and tau in the brain, because, out of the population that displays the relatively high amounts of tau and Aβ, 40% of these over-expressers are cognitively normal functioning individuals [3]. It was concluded that Aβ and tau can be used to diagnose AD, but there are one or more unknown factors that are the key to the onset of AD. This has culminated in attempts being made to discover these unknown factors through analyzing single nucleotide polymorphisms that are associated with cognitive decline [4].

Evaluations of such undiscovered factors which fill in the gaps left by the Aβ and tau hypothesis, the epigenome is the most likely candidate [5]. Epigenetics is defined as heritable changes in gene expression that are, unlike mutations, not attributable to alterations in the sequence of DNA [6]. The long-term aim of the proposed effort is to identify purposeful genomic-based epigenetic biomarkers that can help identify the onset of AD and mild cognitive impairment (MCI) sooner than it is currently achievable [7],[8].

One of the projects seeking to address these undiscovered factors is the epigenetic pipeline of the Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing

Interface (MCI-AD-DTI), started by Professor Leonard Wesley and staffed by his students in the Bioinformatics department of San Jose State University. This project was started when Professor Wesley bore witness to the tedium and time investment required to search through medical data archives for epigenetic information. The MCI-AD-DTI project, upon completion, will drastically increase the efficiency of epigenetics research while maintaining, or even exceeding, the quality and reliability in its results, as though the researcher had done it themselves. The MCI-AD-DTI project is using AD as a model to train and test its results due to the wealth of data available. In addition, AD is being utilized for the potential benefits to the medical community resulting from any insights the MCI-AD-DTI project can discern into the epigenetic nature of AD.

This project seeks to create a Python based Structure-Function Prediction Modules to demonstrate a potential improvement of the MCI-AD-DTI project's protein evaluation code module. The Structure-Function Prediction Modules of this project uses the industry standard protein modeling application Modeller10.4 for its protein modeling qualities. The validation of the new module is performed with a code snippet derived from the existing PyPDB integration with the MCI-AD-DTI project's epigenetic pipeline's protein evaluation module.

Chapter 2

Background

This section gives a detailed description of the motivation for why this project is being built to seek out those proteins whose epigenetic link to AD remains undiscovered, as of yet.

2.1 Alzheimer's Disease Explained

AD is a type of dementia [9] that is a progressive neurodegenerative disorder [10]-[13] and is increasingly recognized as a "synaptic disease" [5]. In addition, AD is characterized holistically by the disintegration of the nervous system [14], and characterized in series by episodic memory impairment [13],[15], temporal and spatial confusion [16], degeneration and then loss of language/attention/planning/reasoning [10]-[12],[16], increasingly severe loss of cognitive function [10]-[13], and finally death [17]. While AD is not the only illness responsible for such symptoms, AD is an epidemic [18] and the most common age-associated neurodegenerative disorder as it is the leading cause of dementia [5],[14],[19],[20]. Specifically, up to 1 in 5 of the community- dwelling older adults aged 65 years and above suffer from MCI [21]. Between 10–15% of patients with MCI may develop dementia each year [22], with the remaining being split among: about 45% of MCI patients maintain stable, 28% progress to AD and 15% return to normal status without recurrence [23]. AD has impacted approximately 47 million or 0.6% of the global population in 2015 [9] with an estimated progression to 75.6 million in 2030 [24],[25], which is extended further to about 1 in 85 individuals over the age of 65 years predicted to be suffering from AD by 2050 [26]. In addition to the human tragedy, AD is already the most expensive disease in the United States at $305 billion/year, which is projected to rise dramatically [27]. Moreover, AD is one of the most well-known diseases in the world due to its

prevalence, the precise cause of the disease is not known, and that the treatments do not modify its progression [1],[13],[28]-[30]. Explicitly, the only approved treatments offer modest symptomatic improvement absent any actual slowing of disease progression [12],[20], and, overall, there is no cure [13]. Given the prevalence and impact of AD, there is a pressing need for development of reliable diagnostic biomarkers that can detect the disease pathology at its incipient stages, i.e., at or even prior to the onset of the ineluctable behavioral and cognitive deficits associated with AD [31].

2.2 Amyloid Hypothesis

The Amyloid hypothesis, which is the hypothesis responsible for the first model used to try and understand AD, is best summarized in the following manner:

> Prior to 1991, the pathogenic mechanisms underlying AD were unknown. This situation changed when mutations in β-amyloid precursor protein (APP) were shown to cause familial autosomal dominant AD (FAD). β-amyloid (Aβ) is the primary constituent of the amyloid plaques characteristic of the disease (reviewed in [11]). Soon after this discovery, mutations in the presenilin-1 and -2 genes were linked to FAD. These genes encode the catalytic subunits of γ-secretase, which generates the fibrillogenic Aβ C-terminus [11]. In carriers of these mutations, amyloid plaques appear first, and the AD that follows resembles the common sporadic form, including tau pathology, inflammation, and cognitive impairment. These observations led to the amyloid hypothesis, which became the focus for therapeutic intervention. [20]

However, despite the genetic and cell biological evidence that supports the amyloid hypothesis, it is becoming clear that AD etiology is complex and that Aβ alone is unable to account for all aspects of AD [4],[13],[21],[32],[33]. For instance, previous work has reported that the amyloid

hypothesis has resulted in false positives and demonstrated a weak to moderate association between Aβ and the degree of cognitive function [21],[34],[35], for between 30% and 40% of normal individuals showed high levels of Aβ and tau [3],[36]. This objection is further supported by the fact that vast overproduction of Aβ peptides in the mouse brain failed to cause neurodegeneration [13] and that recent neuroimaging studies confirm the previous autopsy findings that some AD patients show no amyloid deposits in PET (positron emission tomography) scans [37],[38]. Similarly, repeated failures of clinical trials of anti-Aβ therapies suggest that there may be pathogenic protein independent factors in AD pathogenesis [13],[39],[40]. Two instances demonstrating this are that amyloid clearance with different antibodies was confirmed by PET imaging, but the treatment produced no cognitive benefit [20], and a treatment which reduced Aβ in the cerebrospinal fluid (CSF) by at least 75% failed to preserve cognition [41].

2.3 Importance of Researching Genes

For AD, the molecular pathway associated with it is already characterized by Kyoto Encyclopedia of Genes and Genomes (KEGG) data-base (https://www.genome.jp/pathway/hsa05010) [5]. This pathway contains dozens of genes, such as APP, presenilin 1 (PSEN1), presenilin 2 (PSEN2), and b-secretase 1 (BACE1) to name a few. These genes in turn have important mutations, such as Icelandic APP mutation (A673T) which [1] used to prevent the development of AD during *in vitro* experiments. However, the KEGG AD pathway is not the only approach being taken to investigate the genetic causes of AD. [21] researched the following 15 genes that [4] found to be associated with the glutathione pathway: TMEM14A, RPS16P5, MLIP-IT1, MIR5685, MCM3, LRRC1, GSTA7P, GSTA5, GSTA3, GSTA2, GFRAL, GCM1, FBXO9, FAM83B, and ELOVL5. This resulted in [21] identifying

that the single nucleotide morphism identified as rs10000007 played a significant role toward one of the genes involved in the survival of neurons on ectodermal cells in humans. While already quite diverse, the scientific community has barely scratched the surface upon which genes are indirectly responsible for AD as well as what are the most relevant genetic mutations that either hinder or accelerate the onset of AD.

2.4 Validity of an *in Silico* Approach to Investigating Alzheimer's Disease

In regards to biological experiments it is vital to understand the difference between *in vitro* experiments, *in vivo* experiments, and *in silico* experiments. *In vitro* experiments are those done to a living organism inside a petri dish, which usually lacks the context of how the experiment would function within a healthy and whole variant of the organism. *In vivo* experiments are those done to the whole of living organisms, with the organism outside a petri dish. These can be done in both humans and animals, and these experiments are considered to generate the highest quality data. *In silico* experiments are those done in simulation, with the simulation based upon the real-world data of the medical community's understanding of how the relevant biological processes function. Verification of the potency of *in silico* techniques has been demonstrated on multiple occasions. The first relevant example of *in silico* techniques being utilized is by [1] to calculate possible off-target events for their sgRNA. The results of the *in silico* testing, which stated the only off-target event for their sgRNA, when there were two or fewer mismatches, was a single event isolated to a non coding region [1]. For [1] this was a critical step for their project, as one of the essential factors is to ensure the delivery is safe and efficient. The second example of *in silico* techniques being utilized is by [42]. This occurred when their *in silico* prediction of how Cu(II) would interact with the early AD marker δ-ALA-D, was later *in vivo* corroborated by [43].

2.5 Technical Gap

While the scientific literature on AD is expansive in both the breadth and depth, and the genetic relationships that are thought to be responsible for AD's onset and progression, there still exists a large technical gap in both the existing open questions and those genes and mutations that remain unexamined. The most prominent open question is why the amyloid hypothesis mismatches with real world experiments, as several of the papers have noted AD patients who did not have the Aβ overexpression, and several other papers noted cognitively healthy patients who did have the Aβ overexpression. The concern for the unexamined genes and mutations is instead how complicated the known AD genetic pathways are, a concern further illustrated by even those papers that purport to expand the sphere of knowledge about the AD pathway, as they lament how vast the extent to which the genetic relationships remain unexplored. Thus, the Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI) project seeks to step in as a tool to greatly accelerate the scientific community's advance into the remaining unexplored relationships and hasten towards the answering of the open questions that continue to elude humanity in its endeavors to research and treat this disease.

Chapter 3

Approach

The fundamental approach of this project is to integrate the Modeller10.4 application into a Structure-Function Prediction Modules that emulates the protein evaluating module of the Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI) which is designated as epigen_pipeline_sig_protein module (EPSPM). The intent for the new module is that it can eventually be integrated into the EPSPM and serve to improve the pipeline's ability to investigate the epigenetic relationships underpinning AD.

3.1 Data

In order to construct and debug the Structure-Function Prediction Modules, this project requires a quantity of AD associated wild types, and their associated mutations, that was both sufficiently diverse and that the data was formatted to match the data that was fed into the protein evaluating module of the MCI-AD-DTI project. An example of the minimum viable data set would be three wild types, each with 100 Single Nucleotide Polymorphic (SNP) mutations. The data, that this project utilized, came from the folder created by Frank Kai, one of those responsible for determining the thresholds of the EPSPM [44]. This folder, known as ALL_SNP, contained over 100,000 files of sample SNP mutations and their wildtypes. The creation process for these files was to find the various genes that are labeled as epigenetically linked to AD on the National Center for Biotechnology Information (NCBI), and Frank Kai made a file for each listed SNP mutation, such that the file contained the amino acid sequence of the unique SNP mutation as well as the amino sequence corresponding to the mutant's wild type [44]. This is deemed a sufficient source of data as it covers a prolific quantity of known AD related proteins

and has the requisite diversity of mutations. This provides confidence that the Structure-Function Prediction Modules and its Modeller10.4 application module would be tested with enough edge cases that the threshold of the sorting would be of utmost quality.

3.2 Tools

In regards to the tools, there is: Python, the coding language used in this project; Pycharm, the integrated development environment (IDE) utilized by this project; Salilab, the official Modeller10.4 application's website; and the Modeller application programming interface (API), used to design the aux6help module and aux6mutant module. Python is also the coding language used by the MCI-AD-DTI project, with Pycharm being the first IDE to successfully load the MCI-AD-DTI project's epigenetic pipeline after the Jupyter Notebook IDE failed to do so. The Salilab website served to identify the overall manner in which Modeller10.4 application would be integrated into the Structure-Function Prediction Modules, while the Modeller API was used in the minutia of addressing bugs in the code.

3.3 Algorithms

There are several algorithms used in the code, of which three are the most prominent. First and foremost of the algorithms is that of the Modeller10.4 application, this is an industry standard protein modeling software and the core of this project. In the project Modeller10.4 application takes in a .cif file and a mutant protein sequence and generates 5 different protein models and their Discrete Optimized Protein Energy (DOPE) scores; these scores are then compared against their corresponding wildtype protein sequence to determine whether the mutant preserves the protein functionality of its wildtype, or if it has a change in functionality. The second most important algorithm is Python for the Protein Data Bank (PyPDB) as this library is responsible for generating the .cif files based off of the protein sequences from

ALL_SNP, which are then fed into the Modeller10.4 application's modules of the Structure-Function Prediction Modules. The third algorithm is scoring threshold, which was designed to be an extensible evaluation procedure to be of use beyond AD and SNP mutations.

3.4 Validation

PyPBD, serves as the validation metric in EPSPM, and thus will be utilized for identical purposes in the Structure-Function Prediction Modules. Furthermore, this project will be utilizing confusion matrices for generating the statistics to validate whether Structure-Function Prediction Modules has achieved its goal.

Chapter 4

Method

This section describes the methodology by which the Structure-Function Prediction Modules was constructed and the methodology by which the Structure-Function Prediction Modules evaluates proteins.

4.1 Methodology to Construct the Structure-Function Prediction Modules

The methodology for constructing involves six steps. Step one was acquiring sample data which emulated the files that the epigenetic pipeline of the Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI) generated for its epigen_pipeline_sig_protein module (EPSPM). These files are provided in the ALL_SNP folder from the MCI-AD-DTI project contributor, Frank Kai [44]. The over 100,000 files in the ALL_SNP folder are those that Frank Kai used to determine the thresholds of the EPSPM [44]. Step two was finding how the Modeller10.4 application could be utilized to generate a score by which the likelihood of whether the protein, which was encoded by a mutant sequence, maintained the functionality of the protein encoded by the mutant's wildtype's sequence, could be determined. These are found from the Salilab website, the official website of the Modeller10.4 application. Specifically, the Modeller10.4 application was downloaded from the website following the steps listed, and the specific algorithm utilizing the Modeller10.4 application was derived from a combination of the align2d.py module and the model-single.py module. Step three was creating a Structure-Function Prediction Modules based upon the EPSPM. This Structure-Function Prediction Modules had to be able to interface with both the custom files from the ALL_SNP folder as well as the Modeller10.4 application. Furthermore, the

Structure-Function Prediction Modules needed to emulate the functionality of the EPSPM, such that when it came time to migrate the code from the Structure-Function Prediction Modules and into the EPSPM that the transition went smoothly. Step four was integrating the PDB database into the Structure-Function Prediction Modules. The primary manner in which this was achieved was the use of PyPDB, which was used to generate .cif files, both to feed into the Modeller10.4 application code module, and to serve as a form of validation as found in the EPSPM. Step five was bug testing the Structure-Function Prediction Modules. Through this testing, the most significant thing that was discovered was the need to separate out the Modeller10.4 application code into a separate module. This was done to circumvent a bug where the only way to escape an infinite loop was to have the Modeller10.4 application code be run solely as a subprocess which was checked on a timer to verify whether or not to kill that subprocess. Step six was validating the Structure-Function Prediction Modules with both the ALL_SNP files and the PyPDB implementation (Figure 1).

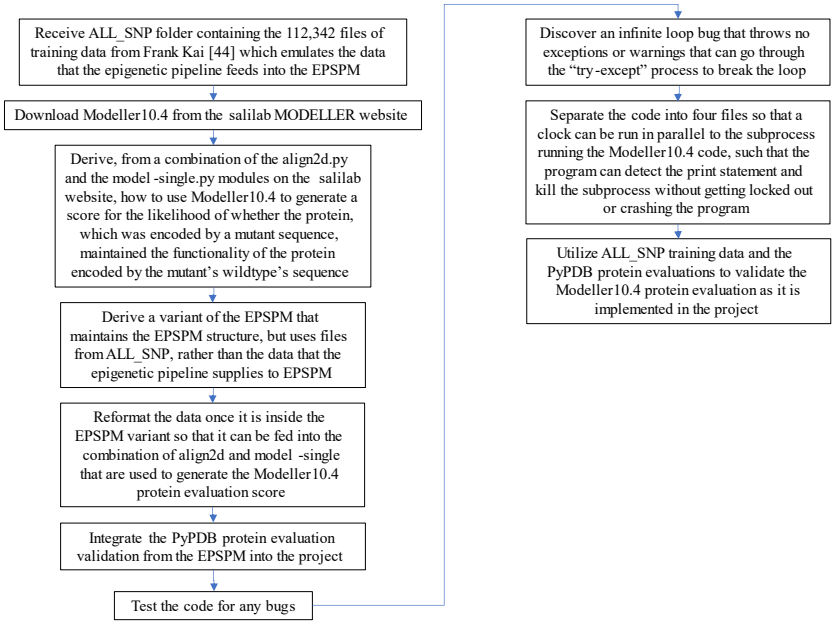

Figure 1: Flowchart of Structure-Function Prediction Modules Construction

4.2 Methodology of the Protein Evaluation

In order for the Structure-Function Prediction Modules to evaluate proteins, the user needs to set up the files as described in Appendix A. Once set up, there are two forms of protein evaluation that are performed in the Structure-Function Prediction Modules. The first is PyPDB version, which is inherited from EPSPM and serves to validate the second method. The second method of generating protein evaluations is the Modeller10.4 version, which generates a set of normalized DOPE scores for the wildtypes and each of their mutants.

The way EPSPM evaluates proteins with PyPDB is replicated in src6 in the pseudo pipeline with the details explained in Appendix B. The way Modeller10.4 evaluates proteins with their DOPE scores involves every module from the Structure-Function Prediction Modules. The src6.py module serves as a structural component as chronicled in Appendix B. The aux6 module is another structural component of the Structure-Function Prediction Modules, but it also serves as section where the normalized DOPE scores of the wildtype and the mutant are evaluated, as recounted in Appendix C. The module aux6help generates the five wildtype DOPE scores and then normalizes them as mentioned in Appendix D, while the aux6mutant module does the same for the mutant as cataloged in Appendix E.

Chapter 5

Results

The following results are derived from a subset of the ALL_SNP folder. This partitioning was done to match those proteins that are referenced in the epigen_pipeline_genes_to_process.dat file. For the files in ALL_SNP of these selected proteins, their wildtype DOPE scores average, median, and standard deviation, as based upon the Modeller10.4 application's module (Table 1).

Table 1: Average, Median and Standard Deviation of the DOPE Scores for the Major Genes

Gene	Average	Median	Standard Deviation
ABCA7	0.811794109503484	0.812632660671874	0.0209432779722801
APOE	0.370899022014451	0.388832738714911	0.0596230130438759
APP	3.14873696379398	3.13720448051296	0.0224422503744293
BIN1	1.04977076225188	1.03852991418138	0.0472459411221466
CD2AP	3.56266334694732	3.55391672523957	0.0422728710520428
CLU	-0.628308838398892	-0.648095416575451	0.119449420601551
CR1	3.55465344726431	3.55310547115527	0.0241804743631271
EPHA1	3.39016564132396	3.3910195609437	0.0187503813507888
INPP5D	1.75344694752547	1.75382810506403	0.00959237179939735
MS4A1	1.75680818890055	1.73350271546339	0.0460174210464811
PICALM	1.23351671718363	1.22172626644825	0.0397182259480443
PSEN1	1.32982008062806	1.27027779414663	0.106508665186033
PSEN2	1.75305830561701	1.79695273490297	0.0948991369830086
SORL1	3.59214412501188	3.58882611515124	0.0229351356177996
TREM2	1.76732268267582	1.7533405971475	0.0896705497654258

5.1 Subsample of Major Proteins from ALL_SNP

In regards to the results of the three sample Single Nucleotide Polymorphic (SNP) mutations from each major protein, the Modeller10.4 application's module corroborated the PyPDB's validation with the following ratios: three SNPs out of three for ABCA7, APOE, CD2AP, and PSEN2; two SNPs out of three for APP, BIN1, CLU, INPP5D, MS4A1, and TREM2; and one SNP out of three for CR1, EPHA1, PICALM, PSEN1, and SORL1. Altogether these proteins have: 27 true positives, 2 false positives, 14 false negatives, and 2 true negatives. This results in the following confusion matrix and statistics (Figure 2 and Table 2).

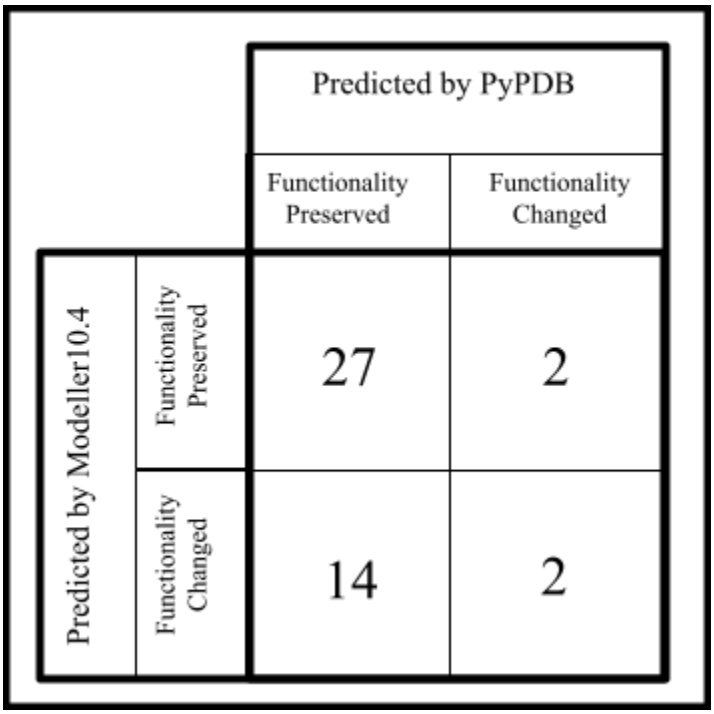

Figure 2: Subsample's Confusion Matrix

Table 2: Statistics of Three Subsamples for each Major Protein

Statistic	Percent
Sensitivity	65.853658536
Specificity	50
Precision	93.103448275
Negative Prediction Value	12.5
Miss Rate	34.146341463
False Positive Rate	50
False Discovery Rate	6.896551724
False Omission Rate	87.5
Positive Likelihood Ratio	131.707317073
Negative Likelihood Ratio	68.292682926
Prevalence Threshold	46.562780185
Critical Success Index	62.790697674
Prevalence	91.111111111
Accuracy	64.444444444
Balanced Accuracy	57.926829268
F1 Score	77.142857142

5.2 APOE

APOE protein is evaluated by both Modeller10.4 and PyPDB when run through the Structure-Function Prediction Modules. The Structure-Function Prediction Modules determined it has: 126 true positives, 114 false positives, 60 false negatives, and 73 true negatives. This results in the following confusion matrix and statistics (Figure 3 and Table 3).

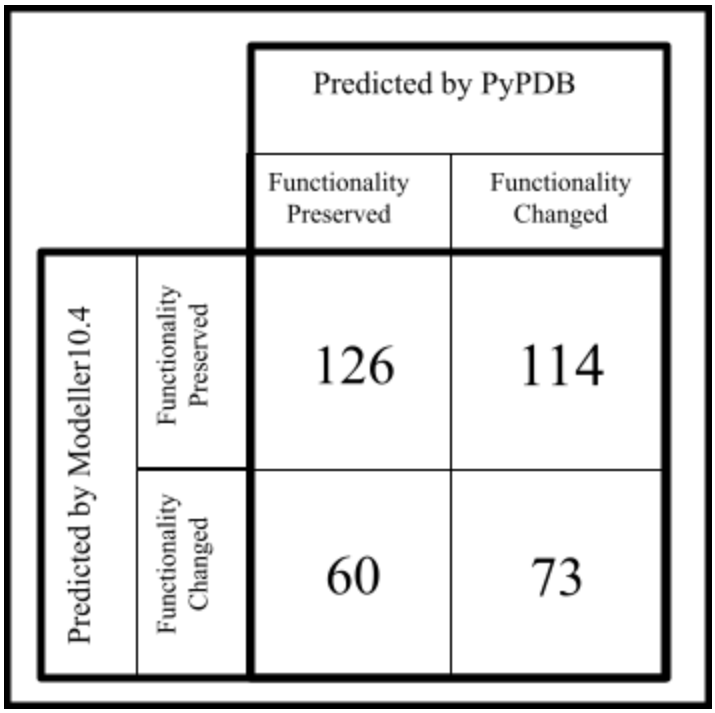

Figure 3: APOE's Confusion Matrix

Table 3: Statistics of APOE

Statistic	Percent
Sensitivity	67.74193548387096
Specificity	39.037433155080214
Precision	52.5
Negative Prediction Value	54.88721804511278
Miss Rate	32.258064516129037
False Positive Rate	60.96256684491979
False Discovery Rate	47.5
False Omission Rate	45.112781954887216
Positive Likelihood Ratio	111.12054329371817
Negative Likelihood Ratio	82.63367211665932
Prevalence Threshold	48.682237711436105
Critical Success Index	42
Prevalence	49.865951742627346
Accuracy	53.35120643431636
Balanced Accuracy	53.38968431947559
F1 Score	59.15492957746479

5.3 BIN1

BIN1 protein is evaluated by both Modeller10.4 and PyPDB when run through the Structure-Function Prediction Modules. The Structure-Function Prediction Modules determined it has: 232 true positives, 1 false positive, 91 false negatives, and 5 true negatives. This results in the following confusion matrix and statistics (Figure 4 and Table 4).

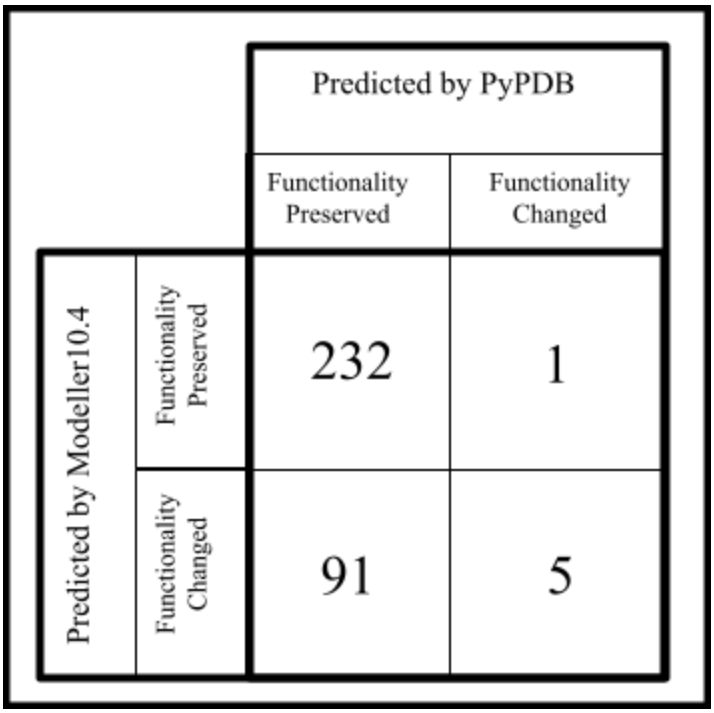

Figure 4: BIN1's Confusion Matrix

Table 4: Statistics of BIN1

Statistic	Percent
Sensitivity	71.826625387
Specificity	83.333333333
Precision	99.57081545
Negative Prediction Value	5.208333333
Miss Rate	28.173374613
False Positive Rate	16.666666666
False Discovery Rate	0.429184549
False Omission Rate	94.791666666
Positive Likelihood Ratio	430.959752322
Negative Likelihood Ratio	0.33808049535
Prevalence Threshold	32.510207246
Critical Success Index	71.604938271
Prevalence	98.176291793
Accuracy	72.036474164
Balanced Accuracy	77.57997936
F1 Score	83.45323741

5.4 PSEN1

PSEN1 protein is evaluated by both Modeller10.4 and PyPDB when run through the Structure-Function Prediction Modules. The Structure-Function Prediction Modules determined it has: 103 true positives, 130 false positives, 94 false negatives, and 117 true negatives. This results in the following confusion matrix and statistics (Figure 5 and Table 5).

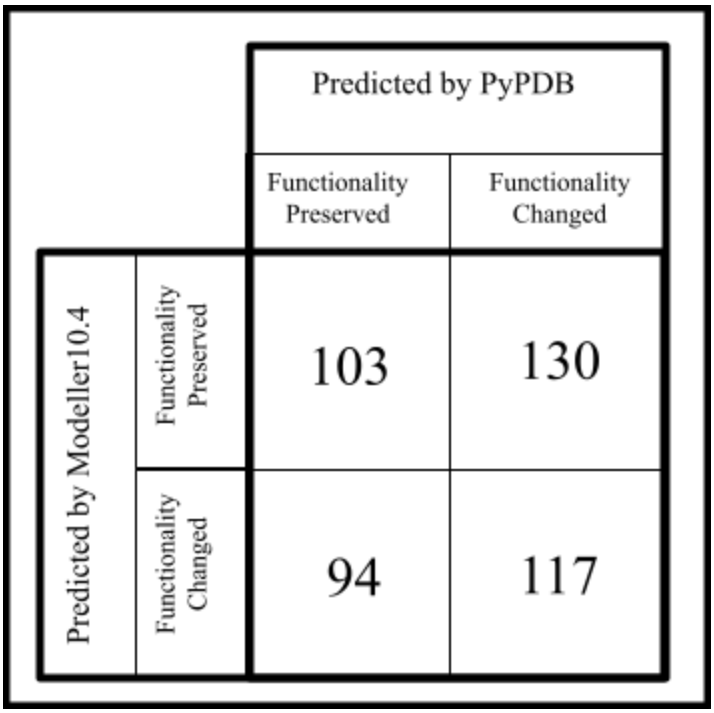

Figure 5: PSEN1's Confusion Matrix

Table 5: Statistics of PSEN1

Statistic	Percent
Sensitivity	52.28426395939086
Specificity	47.368421052631576
Precision	44.206008583690987
Negative Prediction Value	55.45023696682464
Miss Rate	47.715736040609136
False Positive Rate	52.63157894736843
False Discovery Rate	55.79399141630901
False Omission Rate	44.54976303317536
Positive Likelihood Ratio	99.34010152284263
Negative Likelihood Ratio	100.73322053017484
Prevalence Threshold	50.08276060361887
Critical Success Index	31.49847094801223
Prevalence	44.36936936936937
Accuracy	49.54954954954955
Balanced Accuracy	49.826342506011223
F1 Score	47.90697674418605

5.5 PSEN2

PSEN2 protein is evaluated by both Modeller10.4 and PyPDB when run through the Structure-Function Prediction Modules. The Structure-Function Prediction Modules determined it has: 325 true positives, 0 false positives, 81 false negatives, and 1 true negative. This results in the following confusion matrix and statistics (Figure 6 and Table 6).

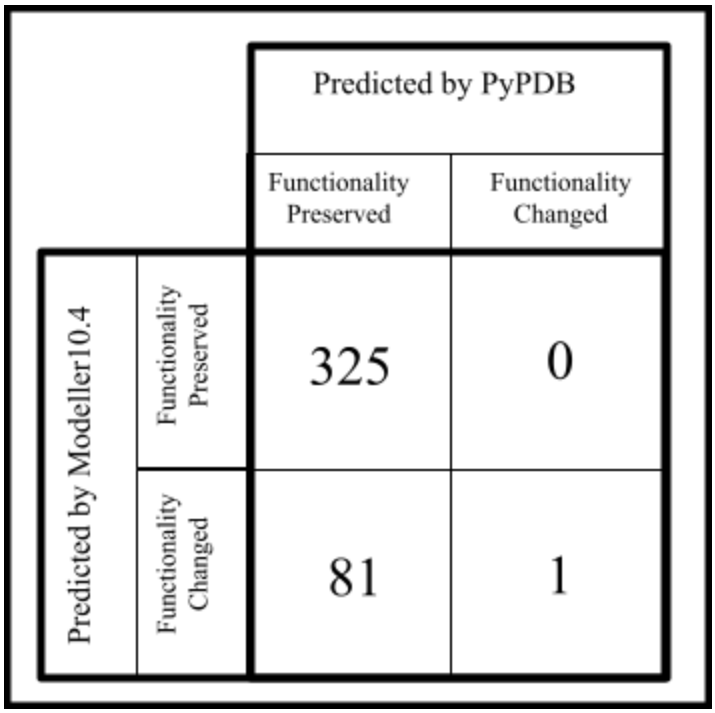

Figure 6: PSEN2's Confusion Matrix

Table 6: Statistics of PSEN2

Statistic	Percent
Sensitivity	80.04926108374384
Specificity	100
Precision	100
Negative Prediction Value	1.2195121951219513
Miss Rate	19.95073891625616
False Positive Rate	0
False Discovery Rate	0
False Omission Rate	98.78048780487805
Positive Likelihood Ratio	not a number due to dividing by 0
Negative Likelihood Ratio	19.95073891625616
Prevalence Threshold	0
Critical Success Index	80.04926108374384
Prevalence	99.75429975429976
Accuracy	80.0982800982801
Balanced Accuracy	90.02463054187192
F1 Score	88.91928864569083

5.6 TREM2

TREM2 protein is evaluated by both Modeller10.4 and PyPDB when run through the Structure-Function Prediction Modules. The Structure-Function Prediction Modules determined it has: 149 true positives, 10 false positives, 32 false negatives, and 1 true negative. This results in the following confusion matrix and statistics (Figure 7 and Table 7).

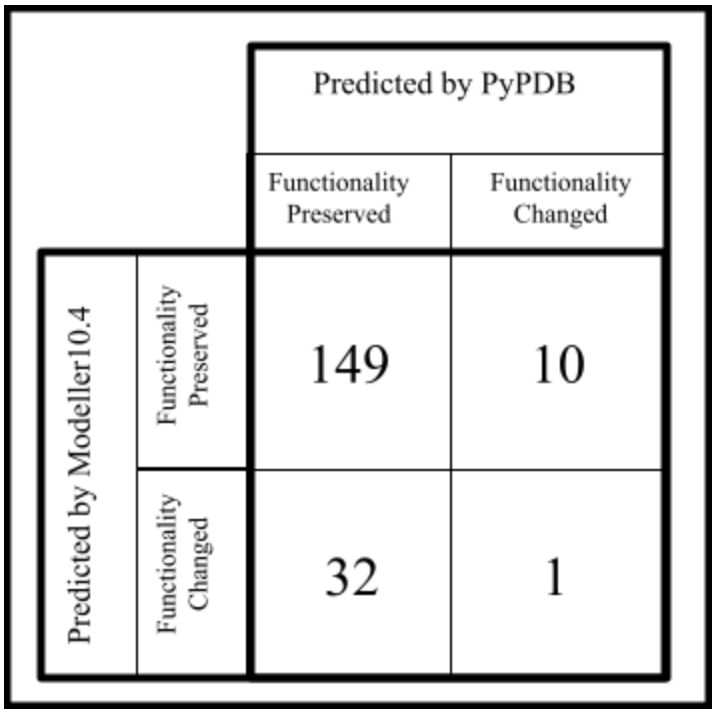

Figure 7: TREM2's Confusion Matrix

Table 7: Statistics of TREM2

Statistic	Percent
Sensitivity	82.320441989
Specificity	9.09090909
Precision	93.710691823
Negative Prediction Value	3.03030303
Miss Rate	17.679558011
False Positive Rate	90.909090909
False Discovery Rate	6.289308176
False Omission Rate	96.969696969
Positive Likelihood Ratio	90.552486187
Negative Likelihood Ratio	194.475138122
Prevalence Threshold	51.240252353
Critical Success Index	78.010471204
Prevalence	94.270833333
Accuracy	78.125
Balanced Accuracy	45.705675539
F1 Score	87.647058823

Chapter 6

Discussion

This chapter discusses the results from chapter 5 and ends with this project's findings from these results.

6.1 Subsample of Major Proteins from ALL_SNP

For the major proteins, as defined by epigen_pipeline_genes_to_process.dat file, the Structure-Function Prediction Modules' favorable results with their percentages: precision is 93.10%, false discovery rate is 6.90%, positive likelihood ratio is 131.71%, negative likelihood ratio is 68.29%, and F1 Score is 77.14%. On the other hand, the Structure-Function Prediction Modules' unfavorable results for those proteins with their percentages: negative prediction value is 12.50%, and false omission rate is 87.50%. The mediocre results with their percentages: sensitivity is 65.85%, specificity is 50.00%, miss rate is 34.15%, false positive rate is 50.00%, prevalence threshold is 46.56%, critical success index is 62.79%, prevalence is 91.11%, accuracy is 64.44%, and balanced accuracy is 57.93%.

6.2 APOE

APOE protein's following statistics are worse than the mean: specificity, precision, false positive rate, false discovery rate, positive likelihood ratio, negative likelihood ratio, critical success index, accuracy, balanced accuracy, and F1 score, with the most extreme divergence coming from a tie between the precision and false discovery rate with a 40.60% difference from the mean for each of them. The worst of these worse scores are: precision, false discovery rate, and F1 score. On the other hand, the statistics that did not change categories, despite their inferiority to the mean, are the following: specificity, false positive rate, positive likelihood ratio, negative likelihood ratio, critical success index, accuracy, and balanced accuracy.

APOE protein's following statistics are better than the mean: sensitivity, negative prediction value, miss rate, and false omission rate, with the most extreme improvement coming from a tie between the negative prediction value and false omission rate with a 42.39% difference from the mean for each of them. All of these statistics improved by a category.

6.3 BIN1

BIN1 protein's following statistics are worse than the mean: negative prediction value and false omission rate. These statistics did not change categories, despite their inferiority to the mean.

BIN1 protein's following statistics are better than the mean: sensitivity, specificity, precision, miss rate, false positive rate, false discovery rate, positive likelihood ratio, negative likelihood ratio, critical success index, accuracy, balanced accuracy, and F1 score, with the most extreme improvement coming from positive likelihood ratio with a 299.25% difference from the mean. The best of these better scores are: sensitivity, specificity, miss rate, false positive rate, critical success index, accuracy, and balanced accuracy. On the other hand, the statistics that change categories, despite their superiority to the mean, are the following: precision, false discovery rate, positive likelihood ratio, negative likelihood ratio, and F1 score.

6.4 PSEN1

PSEN1 protein's following statistics are worse than the mean: sensitivity, specificity, precision, miss rate, false positivity rate, false discovery rate, positive likelihood ratio, negative likelihood ratio, critical success index, accuracy, balanced accuracy, and F1 score, with the most extreme divergence coming from a tie between the precision and false discovery rate with an over 49 percent difference from the mean for each of them. The worst of these worse scores are: precision, false discovery rate, positive likelihood ratio, negative likelihood ratio, critical success index, and F1 score. On the other hand, the statistics that change categories, despite their

inferiority to the mean, are the following: sensitivity, specificity, miss rate, false positivity rate, accuracy, and balanced accuracy.

PSEN1 protein's following statistics are better than the mean: negative prediction value and false omission rate. These statistics both improved categories, when compared to the mean.

6.5 PSEN2

PSEN2 protein's following statistics are worse than the mean: negative prediction value and false omission rate. These statistics did not change categories, despite their inferiority to the mean.

PSEN2 protein's following statistics are better than the mean: sensitivity, specificity, precision, miss rate, false positive rate, false discovery rate, negative likelihood ratio, critical success index, accuracy, balanced accuracy, and F1 score score, with the most extreme improvement coming from a tie between the specificity and false positive rate with a 50 percent difference from the mean for each of them. The best of these better scores are: sensitivity, specificity, miss rate, false positive rate, critical success index, accuracy, balanced accuracy, and F1 score. On the other hand, the statistics that change categories, despite their superiority to the mean, are the following: precision, false discovery rate, and negative likelihood ratio.

6.6 TREM2

TREM2 protein's following statistics are worse than the mean: specificity, negative predictive value, false positive rate, false omission rate, positive likelihood ratio, negative likelihood ratio, and balanced accuracy, with the most extreme divergence being the negative likelihood ratio which was worse by over 126 percent. The worst of these worse scores are: specificity, false positive rate, positive likelihood ratio, and negative likelihood ratio. On the

other hand, the statistics that did not change categories, despite their inferiority to the mean, are the following: negative predictive value, false omission rate, and balanced accuracy.

TREM2 protein's following statistics are better than the mean: sensitivity, precision, miss rate, false discovery rate, critical success index, accuracy, and F1 score, with the most extreme improvement coming from a tie between the miss rate and sensitivity with almost 17 percent difference from the mean for each of them. The best of these better scores are: sensitivity, miss rate, critical success index, and accuracy. On the other hand, the statistics that change categories, despite their superiority to the mean, are the following: precision, false discovery rate, and F1 score.

6.7 Findings

This project's findings with these proteins demonstrate that using Modeller10.4 application to predict the preservation of the functionality of a mutant's protein is a viable endeavor. To be precise, the positive likelihood ratio of greater than one, for the subsampling of the major proteins of ALL_SNP, demonstrates that the algorithm being utilized to evaluate the proteins is working. Although the fact that the algorithm produced a low negative prediction value, and a high false omission rate is concerning.

In relation to the existing Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI) project's epigen_pipeline_sig_protein module (EPSPM) the Structure-Function Prediction Modules has comparable results to the Biopython methods in the EPSPM.

Findings with unexpected values are the prevalence, with its 91.111111111 percent for the major proteins of ALL_SNP, and the likelihood ratios, with its dividing by zero error for the positive likelihood ratio of PSEN2 and its 194.475138122 percent for the negative likelihood

ratio of TREM2. The prevalence demonstrates that the vast majority of the Single Nucleotide Polymorphic (SNP) mutations in ALL_SNP do not actually change the functionality of the protein, as opposed to a more balanced dataset with a prevalence of roughly 50 percent. This was especially notable for PSEN2 which had a single negative sample and resulted in the dividing by zero error as there are no false positives. Whereas the negative likelihood ratio for the TREM2 protein seems to demonstrate that the Structure-Function Prediction Moduless DOPE score algorithm is not optimized for each individual protein.

Chapter 7

Conclusion

In the current project, an effort has been made to construct a Structure-Function Prediction Modules that demonstrates that Modeller10.4 application can be utilized to improve the protein evaluation module of the epigenetic pipeline of the Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI) project. For this purpose to be achieved, the Structure-Function Prediction Modules must successfully emulate the protein evaluation module, integrate Modeller10.4 application in a manner that handles the most common bugs, and deduce an algorithm that evaluates the protein congruous to the validation from the MCI-AD-DTI project's epigenetic pipeline's protein evaluation module.

This project labored under several limitations. The largest imposition was not being able to utilize a copy of the epigenetic pipeline of the MCI-AD-DTI project. This meant that, while the base of the epigen_pipeline_sig_protein module (EPSPM) could be used as a foundation, an entirely new methodology for reading in files to the Structure-Function Prediction Modules had to be created. Furthermore, the Structure-Function Prediction Modules was locked out of the Entrez library utilized for pulling data in the MCI-AD-DTI's epigenetic pipeline.

This project's contribution to society is twofold. The first contribution is its laying of the foundation for integrating Modeller10.4 into the MCI-AD-DTI's epigenetic pipeline, which will serve to enhance the functionality. In turn this will further the MCI-AD-DTI's epigenetic pipeline's capability to resolve the problem with the amyloid hypothesis and potentially discover the true epigenetic relationships underpinning AD. The other contribution is in its enumeration of the process by which Modeller10.4 can be utilized to evaluate proteins to determine if a Single Nucleotide Polymorphic (SNP) mutation has changed or maintained functionality.

Chapter 8

Future Work

For almost every project there is always room for future research and development, and this project is no exception. For instance, this project did not have an opportunity to test the impact of increasing or decreasing the quantity of proteins created by Modeller10.4 application as it generates the DOPE scores, although this quantity is suspected to be an important variable in both runtime as well as the reliability of the scores it generates. Another approach that this project did not have the time and resources to explore is utilizing Modeller10.4 application's family tree algorithm to graph the relatedness between the proteins that the Modeller10.4 application generated for the wildtype and the mutant respectively, as a way to supplement the DOPE scores. Finally, the last major improvement to the project would be integrating the Structure-Function Prediction Modules into the epigen_pipeline_sig_protein module (EPSPM) of the epigenetic pipeline of the Mild Cognitive Impairment and Alzheimer's Disease Detection and Testing Interface (MCI-AD-DTI).

LIST OF REFERENCES

[1] A. Guyon, J. Rousseau, F.-G. Bégin, T. Bertin, G. Lamothe, and J. P. Tremblay, "Base editing strategy for insertion of the A673T mutation in the APP gene to prevent the development of AD *in vitro*," *Molecular Therapy - Nucleic Acids*, vol. 24, pp. 253–263, Jun. 2021, doi: https://doi.org/10.1016/j.omtn.2021.02.032.

[2] Y. Guo *et al.*, "Discordant Alzheimer's neurodegenerative biomarkers and their clinical outcomes," *Annals of clinical and translational neurology*, vol. 7, no. 10, pp. 1996–2009, Sep. 2020, doi: https://doi.org/10.1002/acn3.51196.

[3] M. A. Mintun *et al.*, "[11C]PIB in a nondemented population: Potential antecedent marker of Alzheimer disease," *Neurology*, vol. 67, no. 3, pp. 446–452, Aug. 2006, doi: https://doi.org/10.1212/01.wnl.0000228230.26044.a4.

[4] H.-R. Kim, T. Lee, J. K. Choi, and Y. Jeong, "Genetic variants beyond amyloid and tau associated with cognitive decline," *Neurology*, vol. 95, no. 17, pp. e2366–e2377, Sep. 2020, doi: https://doi.org/10.1212/wnl.0000000000010724.

[5] F. Chen, Q. Guan, Zhi Yu Nie, and L. Jin, "Gene Expression Profile and Functional Analysis of Alzheimer's Disease," *American Journal of Alzheimers Disease and Other Dementias*, vol. 28, no. 7, pp. 693–701, Sep. 2013, doi: https://doi.org/10.1177/1533317513500838.

[6] J. P. Hamilton, "Epigenetics: Principles and Practice," *Digestive Diseases*, vol. 29, no. 2, pp. 130–135, 2011, doi: https://doi.org/10.1159/000323874.

[7] T. Brunette and O. Brock, "Guiding conformation space search with an all-atom energy potential," *Proteins: Structure, Function, and Bioinformatics*, vol. 73, no. 4, pp. 958–972, Jun. 2008, doi: https://doi.org/10.1002/prot.22123.

[8] E. Shakhnovich, “Protein Folding Thermodynamics and Dynamics: Where Physics, Chemistry, and Biology Meet,” *Chemical Reviews*, vol. 106, no. 5, pp. 1559–1588, May 2006, doi: https://doi.org/10.1021/cr040425u.

[9] I. Rawtaer *et al.*, “Early Detection of Mild Cognitive Impairment With In-Home Sensors to Monitor Behavior Patterns in Community-Dwelling Senior Citizens in Singapore: Cross-Sectional Feasibility Study,” *Journal of Medical Internet Research*, vol. 22, no. 5, p. e16854, May 2020, doi: https://doi.org/10.2196/16854.

[10] B. Dubois *et al.*, “Research criteria for the diagnosis of Alzheimer’s disease: revising the NINCDS–ADRDA criteria,” *The Lancet Neurology*, vol. 6, no. 8, pp. 734–746, Aug. 2007, doi: https://doi.org/10.1016/s1474-4422(07)70178-3.

[11] D. J. Selkoe, “Alzheimer’s Disease: Genes, Proteins, and Therapy,” *Physiological Reviews*, vol. 81, no. 2, pp. 741–766, Apr. 2001, doi: https://doi.org/10.1152/physrev.2001.81.2.741.

[12] Ç. Akkaya, S. S. Yavuzer, H. Yavuzer, G. Erkol, M. Bozluolcay, and Y. Dinçer, “DNA damage, DNA susceptibility to oxidation and glutathione redox status in patients with Alzheimer’s disease treated with and without memantine,” *Journal of the Neurological Sciences*, vol. 378, pp. 158–162, Jul. 2017, doi: https://doi.org/10.1016/j.jns.2017.04.051.

[13] S. W. Pimplikar, R. A. Nixon, N. K. Robakis, J. Shen, and L.-H. . Tsai, “Amyloid-Independent Mechanisms in Alzheimer’s Disease Pathogenesis,” *Journal of Neuroscience*, vol. 30, no. 45, pp. 14946–14954, Nov. 2010, doi: https://doi.org/10.1523/jneurosci.4305-10.2010.

[14] M. Citron, “Alzheimer’s disease: strategies for disease modification,” *Nature Reviews Drug Discovery*, vol. 9, no. 5, pp. 387–398, May 2010, doi: https://doi.org/10.1038/nrd2896.

[15] R. A. Sperling *et al.*, “Functional Alterations in Memory Networks in Early Alzheimer’s Disease,” *NeuroMolecular Medicine*, vol. 12, no. 1, pp. 27–43, Jan. 2010, doi: https://doi.org/10.1007/s12017-009-8109-7.

[16] World Health Organization, “Dementia,” *World Health Organization*, 2021. https://www.who.int/news-room/fact-sheets/detail/dementia

[17] C. Reitz and R. Mayeux, “Alzheimer disease: Epidemiology, diagnostic criteria, risk factors and biomarkers,” *Biochemical Pharmacology*, vol. 88, no. 4, pp. 640–651, Apr. 2014, doi: https://doi.org/10.1016/j.bcp.2013.12.024.

[18] Alzheimer's Association, “2017 Alzheimer’s disease facts and figures,” *Alzheimer’s & Dementia*, vol. 13, no. 4, pp. 325–373, Apr. 2017, doi: https://doi.org/10.1016/j.jalz.2017.02.001.

[19] L. C. Lee, M. Q. L. Goh, and E. H. Koo, “Transcriptional regulation of APP by apoE: To boldly go where no isoform has gone before,” *BioEssays : news and reviews in molecular, cellular and developmental biology*, vol. 39, no. 9, p. 10.1002/bies.201700062, Sep. 2017, doi: https://doi.org/10.1002/bies.201700062.

[20] W. J. Ray and V. Buggia-Prevot, “Novel Targets for Alzheimer’s Disease: A View Beyond Amyloid,” *Annual Review of Medicine*, vol. 72, no. 1, Aug. 2020, doi: https://doi.org/10.1146/annurev-med-052919-120219.

[21] K. K. Davis and J. K. Allen, “Identifying cognitive impairment in heart failure: A review of screening measures,” *Heart & Lung*, vol. 42, no. 2, pp. 92–97, Mar. 2013, doi: https://doi.org/10.1016/j.hrtlng.2012.11.003.

[22] C. Hu, D. Yu, X. Sun, M. Zhang, L. Wang, and H. Qin, “The prevalence and progression of mild cognitive impairment among clinic and community populations: a systematic review and meta-analysis,” *International Psychogeriatrics*, vol. 29, no. 10, pp. 1595–1608, Jun. 2017, doi: https://doi.org/10.1017/s1041610217000473.

[23] M. Prince *et al.*, “World Alzheimer Report 2015 The Global Impact of Dementia An AnAlysIs of prevAlence, IncIDence, cosT AnD TrenDs Dr Maëlenn Guerchet Alzheimer’s Disease International,” 2015. Available: https://www.alzint.org/u/WorldAlzheimerReport2015.pdf

[24] A. D. International, “World Alzheimer Report 2019: Attitudes to dementia,” *www.alzint.org*, Sep. 2019, Available: https://www.alzint.org/resource/world-alzheimer-report-2019/

[25] R. Brookmeyer, E. Johnson, K. Ziegler-Graham, and H. M. Arrighi, “Forecasting the global burden of Alzheimer’s disease,” *Alzheimer’s & Dementia*, vol. 3, no. 3, pp. 186–191, Jul. 2007, doi: https://doi.org/10.1016/j.jalz.2007.04.381.

[26] “On the Front Lines: Primary Care Physicians and Alzheimer’s Care in America.” Available: https://www.alz.org/media/Documents/alzheimers-facts-and-figures_1.pdf

[27] C. Ballard, S. Gauthier, A. Corbett, C. Brayne, D. Aarsland, and E. Jones, “Alzheimer’s disease,” *The Lancet*, vol. 377, no. 9770, pp. 1019–1031, Mar. 2011, doi: https://doi.org/10.1016/s0140-6736(10)61349-9.

[28] "Impairments of attention in Alzheimer's disease," *Current Opinion in Psychology*, vol. 29, pp. 41–48, Oct. 2019, doi: https://doi.org/10.1016/j.copsyc.2018.11.002.

[29] N.-H. Trinh, J. Hoblyn, S. Mohanty, and K. Yaffe, "Efficacy of Cholinesterase Inhibitors in the Treatment of Neuropsychiatric Symptoms and Functional Impairment in Alzheimer Disease," *JAMA*, vol. 289, no. 2, p. 210, Jan. 2003, doi: https://doi.org/10.1001/jama.289.2.210.

[30] S. Saharan and P. K. Mandal, "The emerging role of glutathione in Alzheimer's disease," *Journal of Alzheimer's disease: JAD*, vol. 40, no. 3, pp. 519–529, 2014, doi: https://doi.org/10.3233/JAD-132483.

[31] E. S. Musiek and D. M. Holtzman, "Three dimensions of the amyloid hypothesis: time, space and 'wingmen,'" *Nature Neuroscience*, vol. 18, no. 6, pp. 800–806, May 2015, doi: https://doi.org/10.1038/nn.4018.

[32] G. P. Morris, I. A. Clark, and B. Vissel, "Questions concerning the role of amyloid-β in the definition, aetiology and diagnosis of Alzheimer's disease," *Acta neuropathologica*, vol. 136, no. 5, pp. 663–689, 2018, doi: https://doi.org/10.1007/s00401-018-1918-8.

[33] S. Majmudar, "Identify Potential Genetic Variants Associated With Cognitive Impairment and Alzheimer's Disease Beyond B-Amyloid and Tau," Paper, San Jose State University, 2021.

[34] P. Giannakopoulos *et al.*, "Tangle and neuron numbers, but not amyloid load, predict cognitive status in Alzheimer's disease," *Neurology*, vol. 60, no. 9, pp. 1495–1500, May 2003, doi: https://doi.org/10.1212/01.WNL.0000063311.58879.01.

[35] I. Driscoll and J. Troncoso, “Asymptomatic Alzheimers Disease: A Prodrome or a State of Resilience?,” *Current Alzheimer Research*, vol. 8, no. 4, pp. 330–335, Jun. 2011, doi: https://doi.org/10.2174/156720511795745348.

[36] D. A. Bennett *et al.*, “Neuropathology of older persons without cognitive impairment from two community-based studies,” *Neurology*, vol. 66, no. 12, pp. 1837–1844, Jun. 2006, doi: https://doi.org/10.1212/01.wnl.0000219668.47116.e6.

[37] P. Edison *et al.*, “Amyloid, hypometabolism, and cognition in Alzheimer disease: an [11C]PIB and [18F]FDG PET study,” *Neurology*, vol. 68, no. 7, pp. 501–508, Feb. 2007, doi: https://doi.org/10.1212/01.wnl.0000244749.20056.d4.

[38] Y. Li *et al.*, “Regional analysis of FDG and PIB-PET images in normal aging, mild cognitive impairment, and Alzheimer’s disease,” *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 35, no. 12, pp. 2169–2181, Dec. 2008, doi: https://doi.org/10.1007/s00259-008-0833-y.

[39] J. Hardy and B. De Strooper, “Alzheimer’s disease: where next for anti-amyloid therapies?,” *Brain*, vol. 140, no. 4, pp. 853–855, Mar. 2017, doi: https://doi.org/10.1093/brain/awx059.

[40] B. T. Hyman, “Amyloid-Dependent and Amyloid-Independent Stages of Alzheimer Disease,” *Archives of Neurology*, vol. 68, no. 8, p. 1062, Aug. 2011, doi: https://doi.org/10.1001/archneurol.2011.70.

[41] M. F. Egan *et al.*, “Randomized Trial of Verubecestat for Mild-to-Moderate Alzheimer’s Disease,” *New England Journal of Medicine*, vol. 378, no. 18, pp. 1691–1703, May 2018, doi: https://doi.org/10.1056/nejmoa1706441.

[42] Cláudia Vargas Klimaczewski, Pablo Andrei Nogara, Nilda Vargas Barbosa, and J. Batista, “Interaction of metals from group 10 (Ni, Pd, and Pt) and 11 (Cu, Ag, and Au) with human blood δ-ALA-D: *in vitro* and *in silico* studies,” *Environmental Science and Pollution Research*, vol. 25, no. 30, pp. 30557–30566, Sep. 2018, doi: https://doi.org/10.1007/s11356-018-3048-1.

[43] Q. I. Garlet *et al.*, “Delta-Aminolevulinate dehydratase and glutathione peroxidase activity in Alzheimer’s disease: a case-control study,” *EXCLI journal*, vol. 18, pp. 866–875, 2019, doi: https://doi.org/10.17179/excli2019-1749.

[44] Identifying Potential Alzheimer’s Disease Biomarkers Beyond Amyloid-Beta and Tau, Frank Cai, San Jose State University, Computer Science Department Report, December 2024.

Appendix A

Methodology for Running the Structure-Function Prediction Modules

The methodology for running the Structure-Function Prediction Modules is described in the five following steps. Step one is to sort the files into the correct folders. The aux_data folder should contain the ALL_SNP folder and the aux6 .py file, while the aux_src folder should contains aux6help, aux6mutant, and src6 .py files. Step two is removing all .txt files that are inside the aux_data folder, aside from those inside the ALL_SNP folder. Step three is transfering the .txt files from the ALL_SNP folder into the aux_data folder. Step four is running src6 as main, which is primarily done through an IDE like Pycharm. Step five is to interpret the results from the Structure-Function Prediction Modules, which is collected in the output.text file in the aux_src folder.

Appendix B.

src6 Methodology

The methodology of src6 is described in the following ten steps. Step one is that the src6 creates the output file output.text. Step two is src6 reading in all the .txt files from aux_src folder. Step three is where src6 converts their data into a Python dictionary, based on each file's wildtype, while also constructing a list of how many unique wildtypes are amongst the files it has read in. Step four is where src6 loops through the wildtypes and compares their mutations. Step five is when src6 generates a series of .cif files based on the protein sequence of the wildtype. Step six is when src6 generates and formats the seq.ali file to contain the protein sequence of the wildtype for this iteration of the loop from step four. Step seven is when src6 loops through the mutants for this iteration of the step four loop where it both performs the PyPDB comparison emulating the PyPDB comparison of EPSPM and generates the corresponding seqm#.ali files per each mutant. Step eight is when src6 generates the meta files that are being utilized to pass information from one module to another. Step nine is where src6 calls aux6 with the information of the wildtype for this iteration of src6's step four as well as the information of all the mutants that share this iteration's wildtype. Step ten is when src6 deletes the seqm#.ali files from step seven to save memory (Table 8).

Table 8: src6 Process

Step	Files created or removed	Is a substep inside of a src6 loop	Description
One	output.text	No	Readies output
Two	n/a	No	Reads in input files
Three	n/a	No	Process text from input files
Four	n/a	No	Loop through each wildtype
Five	Makes about a dozen .cif based on PyPDB processing the Wildtype	yes (step four)	Use PyPDB to make protein files based on the wildtype
Six	Makes seq.ali version of wildtype sequence	yes (step four)	Convert wildtype sequence to seq.ali in preparation for Modeller10.4
Seven	Makes one seqm#.ali per mutant, where # is the index of the mutant (i.e. seqm1.ali, seqm2.ali, etc.)	yes (step four)	Loop through the mutants for this wildtype and generate their PyPDB score as well their seqm#.ali file in preparation for validating the corresponding Modeller10.4 scores
Eight	Makes snitchM.text, snitchMutant.text, snitchW.text, and snitchWild.text	yes (step four)	Generate the files that aux6 will use to give or receive information from the other modules when other approaches would fail
Nine	See table 2	yes (step four)	Run aux6 with appropriate parameters
Ten	Deletes all seqm#.ali	yes (step four)	Delete the seqm#.ali files for this loop

Appendix C.

aux6 Methodology

The methodology of aux6 can be described in the following twelve steps and occurs within the Structure-Function Prediction Modules as a continuation of src6's step nine. Step one is a loop where aux6 finds a .cif file that is compatible with the wildtype and generates five normalized DOPE scores. Step two checks if all the .cif generated in src6 step six for this wildtype have all been forcefully broken out of their endless loop, if so then it skips to aux6's step six as well as raising a flag for aux6's step seven that this wildtype has no .cif that is compatible with it. Step three both creates the subprocess which is running aux6help, but also starts the clock responsible for killing the subprocess if it needs to be interrupted. Step four checks if aux6help's step five triggered the kill clocks, in which case it moves onto aux6 step one loop's next iteration. Step five only occurs when aux6help's step five successfully generates DOPE scores, and is when aux6 deletes the files generated by aux6help to clear up memory. Step six is where aux6 loops through each mutant and determines whether the Modeller10.4 application demonstrates that a mutant has preserved or changed the functionality from its wildtype. Step seven checks for the flag from aux6's step two, if the flag was triggered then aux6 skips step eight through step twelve as Modeller10.4 application found no .cif that was compatible with the wildtype. Step eight both creates the subprocess which is running aux6mutant, but also starts the clock responsible for killing the subprocess if it needs to be interrupted. Step nine checks if aux6mutant's step five triggered the kill clocks, in which case aux6 notes that this mutation is incompatible with the .cif that aux6 found to be compatible with the wildtype and moves onto step six's next iteration. Step ten only occurs when aux6mutant's step five successfully generates DOPE scores, and is when aux6 deletes the files generated by

aux6mutant to clear up memory. Step eleven checks if one or more of the wildtype's five normalized DOPE scores or the mutant's five normalized DOPE scores is missing, in which case aux6 notes that a DOPE score was missing and it moves onto step six's next iteration. Step twelve analyzes the two sets of DOPE scores to determine Modeller10.4 application's prediction about the mutant, as well as whether this prediction matches with src6's step seven PyPDB's prediction for the mutant (Table 9).

Table 9: aux6 Process

Step	Files created or removed	Is a substep inside of an aux6 loop	Description
One	n/a	no	Loop through the .cif files
Two	n/a	yes (step one)	If out of .cif files move to step six and flags step seven
Three	See table 3	yes (step one)	Start a clock and open aux6help in a separate process
Four	n/a	yes (step one)	If the clock from step four of aux6help and step three of aux6 triggered and killed the process running Modeller10.4, then start the next iteration of aux6 step one's loop
Five	Delete the files from aux6help	yes (step one)	if aux6help successfully generates the five DOPE scores then it deletes the leftover files from aux6help and moves to step six instead on continuing the loop
Six	n/a	no	Loop through the mutations
Seven	n/a	yes (step six)	If the flag from step two was raised, then update output.text that the mutant's wildtype had no valid .cif and skip steps eight through twelve
Eight	See table 4	yes (step six)	Start a clock and open aux6mutant in a separate process
Nine	n/a	yes (step six)	If the clock from step four of aux6mutant and step eight of aux6 triggered and killed the process running Modeller10.4, then update output.text that this mutant doesn't match the wildtype's .cif and start the next iteration of aux6 step one's loop
Ten	Delete the files from aux6mutant	yes (step six)	if aux6mutant successfully generates the five DOPE scores then it deletes the leftover files from aux6mutant
Eleven	n/a	yes (step six)	If a DOPE score is missing update output.text and skip step twelve
Twelve	n/a	yes (step six)	Calculate mean, median, and standard deviation, then update output.text

Appendix D.

aux6help Methodology

The methodology of aux6help can be described in the following six steps and occurs within the Structure-Function Prediction Modules as a continuation of aux6's step three. Step one initiates the Modeller10.4 application's environment. Step two is where all the necessary setup info is appended to the alignment of the protein. Step three is when the salign function of Modeller10.4 application is used on alignment to generate a new .ali file. Step four is when the aux6help clock is started as a counterpart to the clock from aux6's step three, as both of them are required for the successful killing of the subprocess if Modeller10.4 application gets stuck in an endless loop. Step five is when aux6help either successfully generates the DOPE scores or triggers the clock from aux6help's step four to contact the clock from aux6's step three to kill the subprocess that is running aux6help and break the loop. Step six only occurs when aux6help's step five successfully creates the five DOPE scores, and aux6help's step six is where aux6help extracts the normalized DOPE scores and returns them to aux6 (Table 10).

Table 10: aux6help Process

Step	Files created or removed	Is a substep inside of an aux6help loop	Description
One	n/a	no	Setup the parameters needed to run Modeller10.4
Two	n/a	no	Setup the Modeller10.4 environment
Three	makes a .ali file	no	Create an alignment file between wildtype and the .cif it is being compared to
Four	n/a	no	Initiate the kill clock
Five	If successful: makes a .ini, .rsr, and .sch file and five .b#, .d#, and .v# file	no	If caught in an endless loop it triggers the kill clocks. Otherwise, it generates several files and the five DOPE scores
Six	n/a	no	If step five succeeds: normalize the DOPE scores

Appendix E.

aux6mutant Methodology

The methodology of aux6mutant can be described in the following six steps and occurs within the Structure-Function Prediction Modules as a continuation of aux6's step eight. Step one initiates the Modeller10.4 application's environment. Step two is where all the necessary setup info is appended to the alignment of the protein. Step three is when the salign function of Modeller10.4 application is used on alignment to generate a new .ali file. Step four is when the aux6mutant clock is started as a counterpart to the clock from aux6's step eight, as both of them are required for the successful killing of the subprocess if Modeller10.4 application gets stuck in an endless loop. Step five is when aux6mutant either successfully generates the DOPE scores or triggers the clock from aux6mutant's step four to contact the clock from aux6's step eight to kill the subprocess that is running aux6mutant and break the loop. Step six only occurs when auxmutant's step five successfully creates the five DOPE scores, and aux6mutant's step six is where aux6mutant extracts the normalized DOPE scores and returns them to aux6 (Table 11).

Table 11: aux6mutant Process

Step	Files created or removed	Is a substep inside of an aux6mutant loop	Description
One	n/a	no	Setup the parameters needed to run Modeller10.4
Two	n/a	no	Setup the Modeller10.4 environment
Three	makes a .ali file	no	Create an alignment file between wildtype and the .cif it is being compared to
Four	n/a	no	Initiate the kill clock
Five	If successful: makes a .ini, .rsr, and .sch file and five .b#, .d#, and .v# file	no	If caught in an endless loop it triggers the kill clocks. Otherwise, it generates several files and the five DOPE scores
Six	n/a	no	If step five succeeds: normalize the DOPE scores