2021

Augmenting Structure/Function Relationship Analysis with Deep Learning for the Classification of Psychoactive Drug Activity at Class A G Protein-Coupled Receptors

Hannah Willow Shows
Wright State University

AUGMENTING STRUCTURE/FUNCTION RELATIONSHIP ANALYSIS WITH DEEP LEARNING FOR THE CLASSIFICATION OF PSYCHOACTIVE DRUG ACTIVITY AT CLASS A G PROTEIN-COUPLED RECEPTORS

A Dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

by

HANNAH WILLOW SHOWS

B.A., University of Iowa, 2016

2021

Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

3rd December 2021

I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY Hannah Willow Shows ENTITLED Augmenting structure/function relationship analysis with deep learning for the classification of psychoactive drug activity at Class A G protein-coupled receptors BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

_____

Michael Raymer, Ph.D.
Dissertation Director

_____

David Ladle, Ph.D.
Program Director, Biomedical Sciences PhD Program

_____

Barry Milligan, Ph.D.
Vice Provost for Academic Affairs
Dean of the Graduate School

Committee on Final Examination:

_____

David Cool, Ph.D.

_____

Courtney Sulentic, Ph.D.

_____

Jeffery Gearhart, Ph.D.

_____

Michael Leffak, Ph.D.

ABSTRACT

Shows, Hannah Willow. Ph.D., Biomedical Sciences Ph.D. Program, Wright State University, 2021. Augmenting structure/function relationship analysis with deep learning for the classification of psychoactive drug activity at Class A G protein-coupled receptors.

G protein-coupled receptors (GPCRs) initiate intracellular signaling pathways via interaction with external stimuli. [1-5] Despite sharing similar structure and cellular mechanism, GPCRs participate in a uniquely broad range of physiological functions. [6] Due to the size and functional diversity of the GPCR family, these receptors are a major focus for pharmacological applications. [1,7] Current state-of-the-art pharmacology and toxicology research strategies rely on computational methods to efficiently design highly selective, low toxicity compounds. [9], [10] GPCR-targeting therapeutics are associated with low selectivity resulting in increased risk of adverse effects and toxicity. Psychoactive drugs that are active at Class A GPCRs used in the treatment of schizophrenia and other psychiatric disorders display promiscuous binding behavior linked to chronic toxicity and high-risk adverse effects. [16-18]

We hypothesized that using a combination of physiochemical feature engineering with a feedforward neural network, predictive models can be trained for these specific GPCR subgroups that are more efficient and accurate than current state-of-the-art methods.. We combined normal mode analysis with deep learning to create a novel framework for the prediction of Class A GPCR/psychoactive drug interaction activities.

Our deep learning classifier results in high classification accuracy (5-HT F1-score = 0.78; DRD F1-score = 0.93) and achieves a 45% reduction in model training time when structure-based feature selection is applied via guidance from an anisotropic network model (ANM). Additionally, we demonstrate the interpretability and application potential of our framework via evaluation of highly clinically relevant Class A GPCR/psychoactive drug interactions guided by our ANM results and deep learning predictions. Our model offers an increased range of applicability as compared to other methods due to accessible data compatibility requirements and low model complexity. While this model can be applied to a multitude of clinical applications, we have presented strong evidence for the impact of machine learning in the development of novel psychiatric therapeutics with improved safety and tolerability.

Table of Contents

List of Figures

List of Tables

List of Acronyms

5-HT	5-hydroxytryptamine (*also referred to as serotonin*)
AdaBoost	Adaptive boosting
ADHD	Attention-deficit hyperactivity disorder
ANM	Anisotropic network model
ANOVA	Analysis of variance
AUC	Area under the curve
AWS	Amazon Web Services
BLOSUM62	Blocks Substitution Matrix, 62% assumed conservation
ChEMBL	Chemogenomic European Molecular Biology Laboratory Database
CNN	Convolutional neural network
CoV	Coronavirus
DNA	Deoxyribonucleic acid
DRD	Dopamine receptor D (*as in, dopamine receptor D1, D2, etc*)
DSSP	Define secondary structure of proteins
DUD-E	Database of Useful Decoys-Enhanced
ECL	Extracellular loop
ELU	Exponential Linear Unit

ET	Extra trees
FDA	Food and Drug Administration
FN	False negative
FP	False positive
GABA	Gamma-aminobutyric acid
GDP	Guanosine diphosphate
GI	Gastrointestinal
GNM	Gaussian network model
GPCR	G protein-coupled receptor
GPCRdb	G Protein-Coupled Receptor Database
GPU	Graphics processing unit
GTP	Guanosine triphosphate
HIV-1	Human immunodeficiency virus 1
ICL	Intracellular loop
K_d	Dissociation constant
K_i	Inhibition constant
KNN	K-nearest neighbors
LBFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno
LDA	Linear discriminant analysis
LLE	Locally linear embedding
LSD	Lysergic acid diethylamide

MAE	Mean absolute error
MDD	Major depressive disorder
MDPV	Methylenedioxypyrovalerone
MDS	Multi-dimensional scaling
MSE	Mean squared error
NCBI	National Center for Biotechnology Information
nM	Nanomolar
O-PLS	Orthogonal partial least squares (*also referred to as orthogonal projection to latent structures*)
PCA	Principal component analysis
PDB	Protein Data Bank
PDE	Phosphodiesterase
PDSP	Psychoactive Drug Screening Program
PLS	Partial least squares
PPP	Pyrrolidinopropiophenone
PRAUC	Precision-recall area under the curve
PVT	Pyrrolidinopentiothiophenone
QSAR	Quantitative structure-activity relationship
RBF	Radial basis function
ReLU	Rectified Linear Unit
RF	Random forest

RFE	Recursive feature elimination
RMSD	Root-mean-square deviation
ROC	Receiver operating characteristics
SAG	Stochastic average gradient
SBDD	Structure-based drug design
SMILES	Simplified molecular-input line-entry system
SMOTE	Synthetic minority oversampling technique
SQL	Structured query language
SSRI	Selective serotonin reuptake inhibitors
SVM	Support vector machine
TN	True negative
TP	True positive
t-SNE	t-distributed stochastic embedding
USD	United States dollar

Acknowledgments

I am incredibly grateful to my advisor, Dr. Michael Raymer, for the amount of support and guidance he has provided to me over the course of my doctoral work. None of this would be possible without his expertise, patience, and most importantly, his enthusiasm and kindness. Thank you for always believing in me. I would like to acknowledge and thank my committee members (Dr. David Cool, Dr. Jeffery Gearhart, Dr. Courtney Sulentic, Dr. Michael Leffak, and Dr. Mill Miller) for all of their crucial guidance and supervision. I would also like to acknowledge and thank Dr. Mill Miller, Dr. David Ladle, and Karen Luchin with the BMS program for this opportunity. In particular, I am highly indebted to Dr. Mill Miller as his advice, humor, and pointed honesty were instrumental in many challenging moments.

Thank you to all of the role models and teachers throughout this journey – in particular, Dr. Tara Smith, Dr. Brandon Davies, and Dr. Emily Cushing.

I owe so much to my family and friends for their love and support. Thank you for always providing a listening ear, a shoulder to cry on, and food. To my incredible support system of brilliant women, with special recognition to Dr. Xiu Huan Yap, Dr. Sara Seibert, Dr. Devon Price, Dr. Dylan Cutler, Dr. Nikita Wagle, Brooke Kranzler, Christina Dietz, Linda Haviv, Chandler McElrath, Melissa Ward, Erika Aguirre, Kimberly Fiock, Coraline Holquist, Danielle Hausberger, Hollyann Seitz, Jennifer Barrow, and Evelyn Kinne. I would also like to highlight Chris Waker and Clayton Allex-Buckner as both colleagues and friends. Most importantly of all, to Zach Powell—there are no words to accurately describe the depth of my gratitude. Thank you for everything.

Dedication

This dissertation is dedicated to the memory of Lucas Marquardt.

1. Introduction

1.1 Significance and Impact

G protein-coupled receptors (GPCRs) are a family of proteins that interact with extracellular ligands to transduce intracellular signals via binding to a partner heterotrimeric G protein. [1]–[5] There are approximately 800 known GPCRs in the human genome with a diverse range of functions, including roles in neurological, homeostatic, and endocrine pathways. [2], [6] As a result, the GPCR family is a major research focus for clinical and pharmacological applications. [1], [7] Over 30% of FDA-approved compounds target GPCRs. [8] In recent years, structure-based drug design (SBDD) methods have gained popularity due to their increased time- and cost-associated efficiency and ability to produce target-specific, low toxicity compounds. [9], [10] Numerous drugs produced by SBDD are currently available in market; of note, multiple HIV-1 and anti-cancer agents were designed with the aid of computational tools. [11], [12] SBDD methods include machine learning (e.g., regression, clustering, and deep learning) and molecular modeling (e.g., ligand docking)

SBDD tools are ideal for the study of GPCR-targeting ligands. A high proportion of drugs targeting GPCRs are at the upper limit of Lipinski's rule of five in terms of molecular weight and lipophilicity, which correlates to lower selectivity and increased risk of toxicity. [13]–[15] In particular, psychoactive drugs that target Class A GPCRs such as serotonin and dopamine receptors (e.g., antidepressants, antipsychotics, and

mood stabilizer agents) display high rates of chronic toxicity and cross-reactivity as a result of low selectivity. [16]–[18] Existing methods for predicting drug-target activities require a high volume of training data and time. We hypothesize that using a combination of physiochemical feature engineering with a feedforward neural network, predictive models can be trained for these specific GPCR subgroups that are more efficient and accurate than current state-of-the-art methods.

1.2 Specific Aims

Specific Aim 1. To test the performance of current machine learning algorithms for the classification of protein binding behavior and determine the extent to which these architectures can be applied to the classification of Class A GPCR-drug interactions.

Objective: 1a. We will establish a baseline performance for modern classification methods of protein-ligand binding behavior via replicating current models from the recent literature [19]–[22] and optimizing their hyperparameters for the best performance against standard benchmark protein interaction datasets. [23]–[27] **1b.** Optimized models will be tested for accuracy against the binary activity classification of an interaction dataset between Class A GPCRs and their respective drug binding partners to establish an additional baseline for the application potential of machine learning models to Class A GPCR binding behavior.

Specific Aim 2. To assess the degree with which model performance of a Class A GPCR subgroup classifier can be maintained as we reduce time complexity via structure-based feature selection.

Objective: Current predictive models often use fully comprehensive structure and/or sequence data in their feature sets risking the inclusion of irrelevant features. Irrelevant additional features may decrease model accuracy and increase computation time. [28] We hypothesize that by rationally decreasing our feature set by focusing on regions of the

structural landscape with the highest variation within a dataset, we will be able to reduce data noise and model time performance. A combination of anisotropic network model analysis and machine learning classifiers will be used to predict the subgroup of Class A GPCR protein structures. We hypothesize that reducing our feature space will decrease average training time of a model without a consequential decrease in classification accuracy. If accuracy is maintained, the resulting decreased computation time of our model will therefore be better suited to examine our data in finer detail and increase our ability to extract a higher degree of learning from our model.

Specific Aim 3. To implement a novel deep neural network classifier for the binary prediction of drug binding activity for a dataset of (a) serotonin and (b) dopamine Class A GPCRs.

Objective: We will develop a deep learning model for the binary activity (active/inactive) classification of Class A GPCR-small molecule interactions. This network will be separately trained and optimized for two subgroups of the Class A GPCRs respectively, the **(3a)** serotonin (5-HT) and **(3b)** dopamine (DRD) receptors. These interactions are clinically relevant to the development of numerous psychoactive compounds; of note, antipsychotic and antidepressant drugs. [8], [17], [27], [29], [30]

Overall Impact: We hypothesize that by implementing modern state-of-the-art strategies to optimize parameters for these models, we will be able to improve upon current drug-target interaction prediction accuracy while establishing a novel baseline performance for a specific clinical application. As a result, an efficacious predictive tool for the binding behavior of these receptor subtypes has considerable impact potential

from a translational perspective, as well as providing a technical innovation for the use of deep learning in computational biochemistry.

2. Literature Review

2.1 G protein-coupled receptors.

The G protein-coupled receptor (GPCR) superfamily of proteins interact with extracellular ligands to influence intracellular G protein-mediated signaling behavior. GPCRs are membrane-embedded proteins structurally characterized by their alpha helix barrel structure. (**Figure 1**) [2], [31]

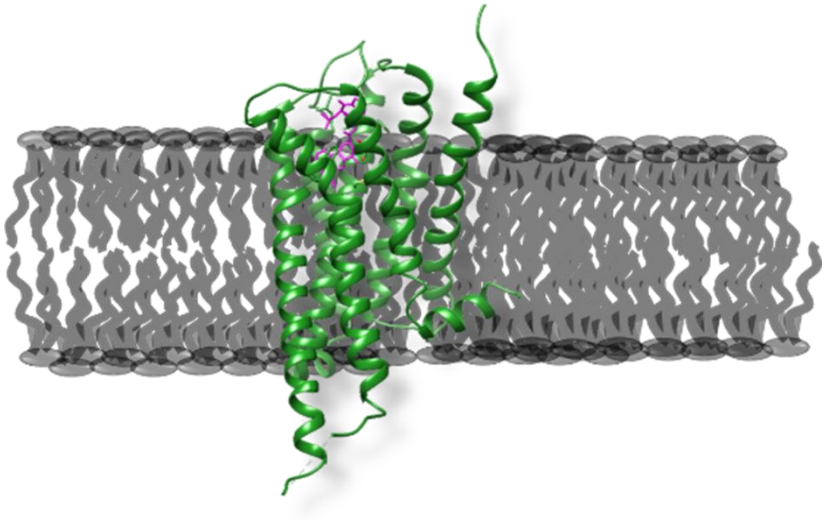


Figure 1. **Schematic of a ligand-bound Class A GPCR embedded in the plasma membrane.** Three-dimensional structure of the human D2-receptor, (green) a GPCR belonging to the A17 subfamily, bound to bromocriptine (purple) in the extracellular orthosteric binding site while embedded in the plasma membrane. Adapted from PDB 6VMS and modeled in UCSF Chimera.

When activated by a ligand, the receptor binds to a partner heterotrimeric G protein and promotes exchange of GTP for GDP, leading to a wide range of downstream signaling effects. [1] The interaction between the GPCR and partner G protein relies on a substantial conformational change of the GPCR resulting from interactions with the

extracellular ligand. Generally, the inactive GPCR conformation stabilizes interactions with a GDP-bound G protein while active conformations promote exchange of GTP for GDP from the G protein alpha subunit. Consequently, the GTP-bound alpha subunit dissociates from the beta and gamma subunits (which also later dissociate) and the dissociated subunits separately mediate downstream signaling activities. [32] This cycle is represented in **Figure 2.**

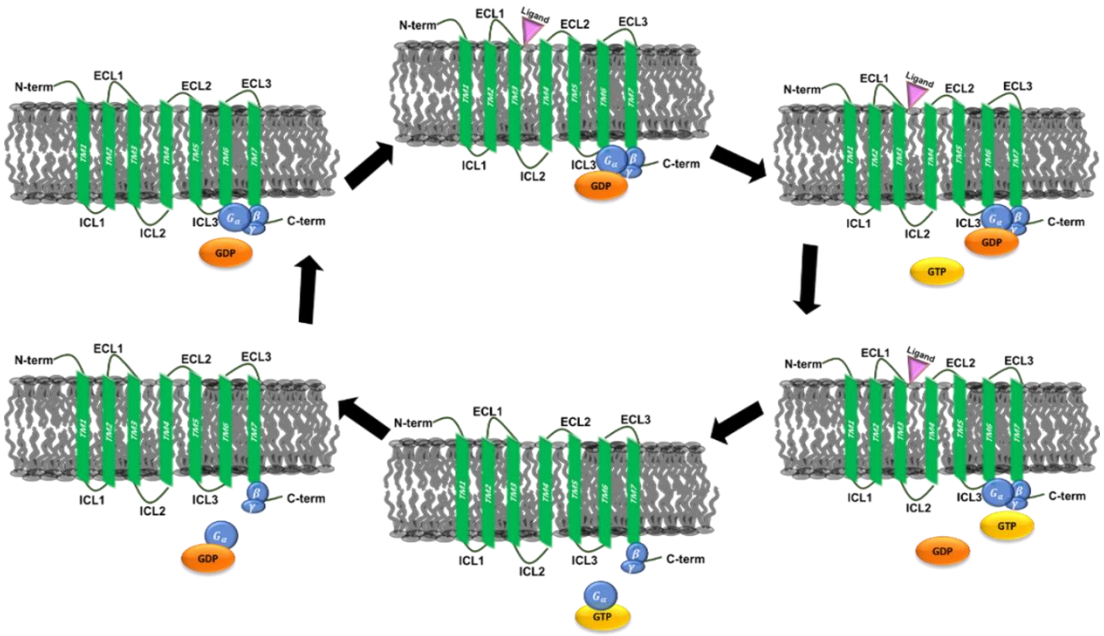


Figure 2. **GPCR activation cycle.** Cycle demonstrating the exchange of GDP for GTP with a partner G protein upon ligand binding to a GPCR.

Agonists, or activating ligands, stabilize the active conformation while inverse agonists stabilize the inactive conformation. Antagonists do not affect the equilibrium of the receptor but instead compete with agonists for the orthosteric binding site. However, GPCRs are not simple on/off switches. There is a preexisting equilibrium between inactive and active conformations, and the equilibrium can be shifted when agonists promote GTP exchange activity above the basal level to favor an active conformation or when inverse agonists lower activity below the basal level and thus favor an inactive

conformation. [5] Partial agonists may affect the activity equilibrium of the receptor at a partial efficacy compared to a full or inverse agonist.

GPCRs are the largest family of membrane proteins and mediate a diverse range of biological processes. GPCR activity is mediated by light, hormones, neurotransmitters, odors, pheromones, and many other peptides and small ligands. [1] While GPCRs share similar structure, they are grouped into Classes A-F (**Figure 3**) based on their ligand binding behavior, with Class A (rhodopsin- and rhodopsin-like receptors) containing 85% of all GPCRs. As a result, Class A is further divided into subgroups A1-A19. [3], [4], [33] Approximately one-third of all FDA-approved drugs target GPCRs, predominately focused on Class A receptors. [34]–[36] The global sales volume for GPCR-targeting drugs was approximately 180 billion USD in 2018. [34]

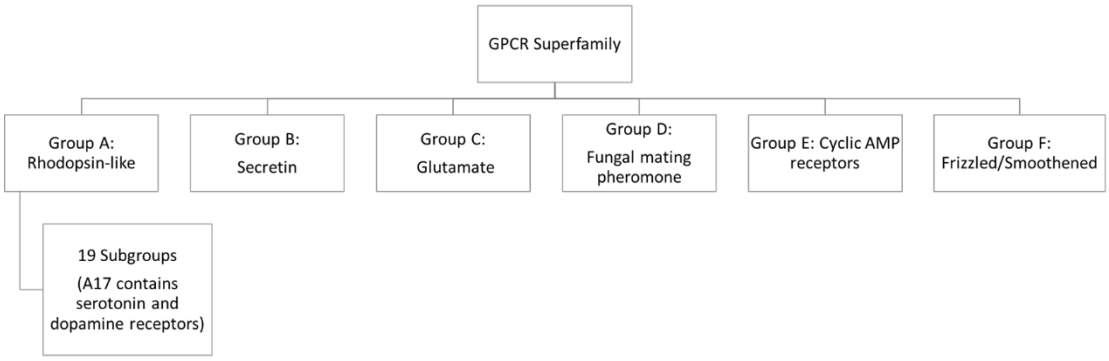


Figure 3. **Classification of the GPCR superfamily.** Classes A-F grouping of the GPCR superfamily. Of note, Class A17 contains the majority of serotonin and dopamine receptors along with the adrenergic, trace amine, and histamine H_2 receptors.

GPCRs possess seven transmembrane-spanning helices, as well as a varying number of extracellular loops (ECL) and intracellular loops (ICL) which interact with extracellular ligands and a G protein, respectively. (**Figure 4)** GPCR ECL contain conserved cysteine residues that form structurally-important disulfide bonds. These ECL

are also typically glycosylated. The ECL2 region is often involved in the binding mechanism within Class A, as well as key transmembrane domain residues (often, a highly conserved tryptophan residue referred to as the toggle switch at position W6.48) and residues within ICL2 and ICL3. [1], [5]

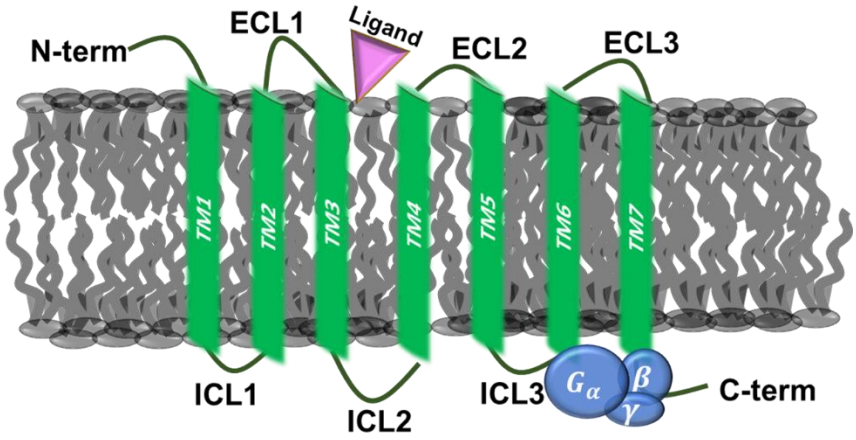


Figure 4. **GPCR binding sites for extracellular ligands and intracellular G proteins.** Two-dimensional representation of a GPCR (green) bound to an extracellular ligand (purple) and an intracellular heterotrimeric G protein (blue).

GPCRs are closely involved in behavior and mood regulation via binding ligands such as serotonin, dopamine, GABA, histamine, glutamate, opioids and cannabinoids. [27], [37]–[41] As a result, they are often studied in the context of behavioral medicine, psychiatry, and substance use disorders. The relationship of certain GPCRs with therapeutic and recreational use of psychoactive compounds has been well characterized. [39], [42]–[46]

2.2 Psychiatric disorder epidemiology and pharmacology.

Therapeutics that target small molecule neurotransmitter (e.g., serotonin, dopamine, norepinephrine) receptors and transporters are often used in the treatment of disorders such as major depressive disorder, bipolar disorder, and schizophrenia. Major depressive disorder (MDD) is characterized by depressed mood, loss of pleasure, weight fluctuation, insomnia or hypersomnia, psychomotor agitation or retardation, fatigue, low self-esteem, and general executive dysfunction. [47] More than 262 million people have MDD, making the disorder the leading cause of disability worldwide. [48] The risk of suicide within MDD-affected individuals is approximately 20 times higher than that of the general population. Additionally, approximately 17% of individuals with MDD have a co-morbid substance use disorder. [49], [50] The literature predominately attributes the pathophysiology of MDD to hypoactivity of monoamine neurotransmitters such as serotonin, although numerous other pathways have been implicated (e.g., cytokine function and the cortico-limbic system). [27], [51], [52]

Selective serotonin reuptake inhibitors (SSRIs) are used to treat depressive symptoms in MDD and other behavioral health disorders via increasing the extracellular concentration of serotonin in the synaptic cleft by limiting its reuptake into the presynaptic cell. [53] While SSRIs are designed to target serotonin transporters, they also often have affinity for norepinephrine and dopamine transporters as well as GPCRs such as the serotonin, opiate, histamine, acetylcholine, and norepinephrine receptors. [42],

[45], [53] While newer generations of antidepressant medication exhibit increased selectivity, the SSRI class of drugs exhibit side effects such as sexual dysfunction, increased risk of bone fractures, suicide risk, akathisia, weight gain, QT prolongation and interactions with anticoagulants that result in a risk of increased GI bleeding. [42], [54]–[59] Examples of common SSRI drugs include citalopram, escitalopram, fluoxetine, fluvoxamine, paroxetine, and sertraline.

Bipolar disorders are characterized by recurrent, cyclic episodes of elevated mood and depression which cause characteristic cognitive, physical, and behavioral symptoms. [60] Hallmark symptoms of bipolar disorder include alternating states of mania and depression, impaired executive function, and psychosis. [61]–[63] Over 30% of individuals with bipolar also struggle with a substance use disorder and are twice as likely to experience physical or sexual abuse compared to neurotypical individuals. [64], [65] Greater than 6% of bipolar individuals die via suicide, highlighting the necessity for effective symptom management. [66], [67] Lifelong psychiatric treatment is often necessary for bipolar patients; however, large meta-analysis studies assessing adherence to medication regimen found that only 54% of bipolar patients were fully adherent to maintenance drugs. [41], [68], [69] Bipolar disorder is often treated with antipsychotic or benzodiazepine medications with known side effects such as narcolepsy or insomnia, weight gain or loss, tremors, akathisia, parkinsonism, and dystonia. [16], [17], [43], [69]

Schizophrenia and schizoaffective disorders are characterized by hallucinations, delusions, and disordered thinking that impairs daily functioning. [40] Many of the cognitive defects present in bipolar disorder are also symptoms of schizophrenia, as well as overlapping psychotic symptoms (although these symptoms are typically more

prominent in schizophrenia). [70], [71] Approximately 50% of schizophrenic individuals have mental and/or behavioral health comorbidities; of note, the rate of substance use disorders is twice as high in the schizophrenic population as compared to the general population. [44], [72] An estimated 5% of people with schizophrenia die via suicide. [66] Schizophrenic patients are often prescribed similar medications to those being treated for bipolar disorder and experience comparable prescription noncompliance rates as bipolar patients. [40]

There are 114 known specific loci in the human genome that contribute to the risk of both schizophrenia and bipolar disorder, compared to only four known regions that differentiate the biology of the two disorders. [73] While the molecular basis of these diseases is not fully understood, the two disorders both demonstrate defects with dopamine signaling pathways. Antipsychotic medications are a subclass of neuroleptic psychoactive compounds used to treat a range of psychiatric disorders primarily known for their efficacy in the treatment of schizophrenia and bipolar disorder. Antipsychotics are grouped into the typical and atypical classes based on chemical similarity. Typical antipsychotics, also known as first-generation antipsychotics, include the butyrophenones, diphenylbutylpiperidines, phenothiazines and thioxanthenes. Atypical antipsychotics, or second-generation antipsychotics, include the benzamides, benzisoxazoles, phenylpiperazines, and tricyclics. Antipsychotic medications often function as dopamine receptor antagonists (in particular, DRD2) and serotonin receptor antagonists (in particular, 5-HT2A and 5-HT2C) but do not have a selective mechanism. [16], [74], [75] In addition to binding a wide range of dopamine and serotonin receptors, antipsychotics also can interact with multiple other GPCRs including the cholinergic and

histaminergic receptors. [17], [30], [76], [77] The number of target receptors a drug interacts with is directly correlated with its adverse effects. [78], [79] Side effects are often reported as a factor in poor prescription adherence in both bipolar and schizophrenia patients due to negative impact on quality of life. [17], [53], [68] Antipsychotics have documented side effects including sexual dysfunction, metabolic syndrome, extreme weight fluctuation, gynecomastia, movement disorders, renal toxicity and shrinkage of brain tissue volume leading to permanent disability. [16]–[18], [80]

Collectively, this clinical and biochemical evidence demonstrates a need for a thorough investigation into the pharmacodynamics of GPCR interactions with psychoactive drugs, allowing for the development of future candidates that are capable of highly selective and efficacious receptor antagonism.

2.3 Pharmacodynamics and SBDD.

The pharmacodynamics of a drug-target interaction include the study of how a drug affects an organism on a molecular and physiological level. Studying the chemical interactions that result in receptor binding on a molecular level often help explain the dose response, toxicology, and physiologic changes that result from a drug interaction on the organismal level. [81] As a result, structural biochemistry analysis of protein-ligand interactions yields useful information for studying the behavior of therapeutic compounds. [7], [82], [83] SBDD has gained popularity over the last thirty years due to resulting production of selective, lower toxicity compounds associated with decreased time- and financial- cost. Several FDA-approved drugs were produced by SBDD methods, including successful HIV-1 inhibitors and chemotherapy agents developed using computational tools. [11], [84] SBDD methodology may include computational techniques such as machine learning (e.g., regression, clustering, and deep learning techniques) and molecular modeling (e.g., ligand docking, molecular dynamics trajectories).

2.4 Computational tools.

2.4.1 Modeling and simulation

In early stages of the integration of *in silico* techniques into biochemical research, simple statistics-based simulation models (such as Monte Carlo sampling) were used to represent atomic positions within a molecule over a time course. [85]–[89] These simulations evolved further into complex molecular dynamics trajectories capable of predicting and estimating the thermodynamics and kinetics of a molecule's intermolecular interactions with binding partners or solvent, as well as its intramolecular interactions. [82], [83], [90] Advances in molecular modeling technology have been crucial for the development of modern structure determination techniques and ligand docking models have become a mainstay of pharmacological research. (**Figure 5**) [91], [92]

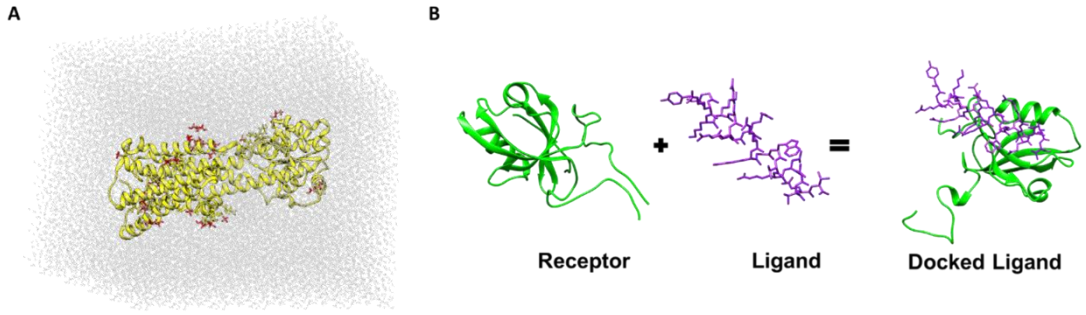


Figure 5. Molecular modeling and dynamics simulations strategies for *in silico* structural analysis. A Simulated model of a protein structure (yellow) bound to multiple ligands (red) while solvated in NaCl solution. Generated using UCSF Chimera. B Cartoon representation of ligand docking in which a protein receptor model (green) is simulated in solvent (not shown) with a ligand (purple; not to scale) to generate a predicted bound structure.

While modeling and simulation-based techniques are powerful, they possess a complex feature space and as a result, are computationally expensive. [93], [94] Output quality is often sacrificed in the form of short timescales and oversimplified representations of the physiochemical parameters to minimize the feature space, allowing for calculations to be completed in a reasonable timeframe. [95]–[97] Within recent years, machine learning tools capable of handling “big data” problems have been applied to computational biochemistry research to address the large feature space and amount of output data produced when studying molecular interactions. [20], [22], [98]–[100]

2.4.2 Machine learning

Machine learning models are named for their ability to learn from data. Machine learning is defined with respect to a given task as an algorithm in which performance improves with experience. [101] Machine learning models are shown examples of data observations to learn from where our data is described by a collection of features. Each data observation is typically represented as a vector $\boldsymbol{x}$ where each entry x_i of the vector is a unique feature. (**Figure 6**) Common machine learning tasks include classification, regression, anomaly detection, and density estimation. [102]

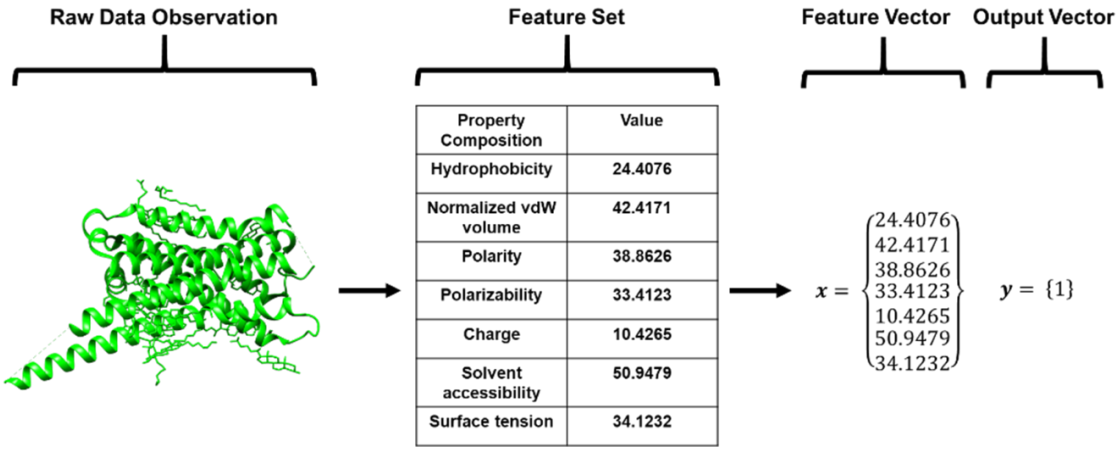


Figure 6. Feature extraction process from a protein structure. As an example, consider a protein structure as a single raw data observation. Descriptors are extracted from this observation, such as polarity and charge. These descriptor values can be stored as feature values in a given vector index x_i. This input vector of features x can be mapped to an output value or vector of values y such as the binary output '1' shown here, such as a classifier predicting a drug interaction being 'active' or 'inactive' for the given receptor, which can be coded as {1} or {0}.

When machine learning models are given example data to learn from, we refer to this process as training and the data as the training dataset. The model is categorized as

supervised or unsupervised based on the presence of labels in our data. Unsupervised models observe examples of data in the form of a random vector $\boldsymbol{x}$ and attempt to estimate the probability distribution $p(\boldsymbol{x})$. Supervised models observe examples of data in the form of a random vector $\boldsymbol{x}$ and an associated vector $\boldsymbol{y}$ (label or output data) and learn to predict $\boldsymbol{y}$ from $\boldsymbol{x}$ by estimating $p(\boldsymbol{y} \mid \boldsymbol{x})$. Unsupervised models train on an unlabeled dataset containing many features and identify useful structural patterns in the data based on these features. Supervised models train on labeled datasets where the relationship between the input and output can be estimated. [102]

2.4.3 Performance Metrics

We evaluate performance measures of our model on a test set of data. Our test dataset contains example data that the model has not previously seen. Performance is typically assessed via measures of accuracy, precision, and error. One must choose the most appropriately informative performance measure for a given specific task (e.g., measuring the performance of a classifier versus a regression model). Common measures of accuracy and error for a classifier model, such as the models presented in this research, include mean absolute error (MAE), mean squared error (MSE), classification accuracy, confusion matrices, F1 score, and other measures of precision and recall such as the receiver operating characteristics (ROC), area under curve (AUC), and area under precision-recall curve (PRAUC). [103]–[105]

MAE refers to the average difference between the true and predicted values for a dataset. MAE is a good measure of the magnitude of difference between predicted and true output but does not indicate the direction of the error (e.g., overpredicting one class over the others). [105] MAE is represented by:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| \qquad (1)$$

MSE is very similar to MAE, mathematically differing only by taking the average of the squared difference between the predicted and true values for a dataset. [106] In

practice, this allows the gradient to be computed more easily. By taking the square of the difference there is a much greater penalty for larger errors. [105] MSE is defined as:

$$MSE = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2 \qquad (2)$$

Classification accuracy is a much simpler measure of accuracy; however, it is not as descriptive when there is an uneven distribution of samples between classes.

$$Classification\ Accuracy = \frac{Total\ number\ of\ correct\ predictions}{Total\ predictions\ made} \qquad (3)$$

A confusion matrix describes the complete performance of a given classifier with the counts of true positives, false positives, true negatives, and false negatives. These counts form the basis for other performance measures such as precision, recall, F1-scoring and AUC/ROC curves. [104], [105] When testing a model with n samples, the confusion matrix resembles:

n = TN + FP + FN+ TP	Predicted No	Predicted Yes
No	Total number of true negatives (TN)	Total number of false positives (FP)
Yes	Total number of false negatives (FN)	Total number of true positives (TP)

Precision and recall are the foundation of many other metrics, defined as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \qquad (4)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \qquad (5)$$

Precision represents the fraction of relevant instances to all retrieved instances, while recall represents the fraction of retrieved instances to all relevant instances. Precision measures the ability of a model to return *only* relevant instances while recall (also referred to as sensitivity) measures the ability of a model to identify *all* relevant instances. In some cases, we may optimize precision or recall at the expense of the other or we may attempt to find an optimal balance between the two metrics. The F1 score [105] is the harmonic mean of precision and recall defined by:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \qquad (6)$$

Harmonic mean is used in place of an arithmetic mean because the harmonic mean strongly penalizes extreme values (e.g., with precision = 1 and recall = 0, the arithmetic mean is equal to 0.5 but the F1 score is 0). The F1 score ranges from 0 to 1 with higher performing models having greater F1 scores.

The ROC curve is a probability curve plotted with the true positive rate against the false positive rate, providing information about the model performance for binary classification at various decision threshold settings. The AUC measures the area under the ROC curve ranging from 0 to 1. When AUC is close to 0, the model is reciprocating the result (i.e., predicting classes as the opposite values). When AUC is close to 1, the

model has very good measure of separability and accuracy. When AUC is 0.5, the model has no class separation capacity at all. [107]

The precision-recall curve is a probability curve plotting the recall against the precision for various decision threshold settings. The PRAUC refers to the area under the precision-recall curve. As a result of recall increasing, precision decreases. PRAUC metrics can be useful in determining the optimal threshold for precision/recall balance. [108]

When building a binary classification model, it is important to consider multiple accuracy metrics to determine the decision threshold. The F1-score is an important metric in most binary classification problems; in particular, models where the positive class is more important. When data is relatively balanced between classes, the ROC AUC metric is an appropriate measure of model accuracy. However, when data is heavily imbalanced, the PRAUC is more appropriate. [105], [109], [110]

Threshold is one of the many important hyperparameters that must be tuned for a machine learning model. Hyperparameters refer to parameters whose values are determined prior to the learning process (e.g., threshold, learning rate, maximum number of features) in contrast to parameters determined via training (e.g., node weights). Each specific machine learning algorithm requires unique hyperparameters to be tuned to improve the speed and quality of the learning process. Performance measures for training and test error values guide the hyperparameter tuning process. [102]

Random search and grid search are simple and commonly used search optimization algorithms. In random search, the search space (or volume to be searched where each

dimension represents a hyperparameter and each observation represents one possible set of hyperparameters) is a bounded domain of possible hyperparameter values and combinations are randomly sampled in that domain. Grid search, in contrast, evaluates every single possible combination in the domain.

Training error refers to any measure of error (i.e., proportion of incorrectly predicted values) for the training set. The test error, or generalization error, refers to the expected value of the error on a previously unseen dataset. When optimizing the performance of a machine learning model, we want to both minimize our training error and the gap between the training and test errors. When the training error is too large, it results in underfitting of the data. When the magnitude of difference between the training and test error is too large, overfitting occurs. [102]

In supervised machine learning models, the error should be minimized while learning from training data. The loss function (i.e., error function) quantifies the associated error for a single training example. The cost function, sometimes used synonymously with the loss function, describes the average loss over the entire training set. Optimization strategies such as gradient descent aim to minimize this cost function. In addition to the importance of optimizing a model, regularization of the model should also be considered. Regularization refers to any modification that is made to an algorithm with the intended purpose of reducing the test error but not the training error. If the training data contains too much noise, there is a risk of overfitting the data and the model will not generalize well to future dataset testing. [102] Often, the regularizer is in the form of a penalty added to the loss function. Regularization effectively discourages overly flexible models and the resulting risk of overfitting. To successfully achieve this

goal, a regularizer should reduce model variance without increasing bias (i.e., our model should be generalizable to previously unseen data in a test dataset without overly increasing error due to false assumptions). [111] The variance of an estimator θ is equivalent to the square of the standard error of θ where:

$$SE(\hat{\mu}_m) = \frac{\sigma}{\sqrt{m}} \qquad (7)$$

while the bias of an estimator θ is defined as $bias(\hat{\theta}_m) = \mathbb{E}(\hat{\theta}_m) - \theta$, where the expectation $\mathbb{E}(\hat{\theta}_m)$ is over samples from a random variable and θ is the true value of θ. [105]

Common examples of regularization include applying an L1 or L2 vector norm penalty to the network optimization such as the use of Ridge regression in linear models or penalizing a neural network loss function for large weights. The L1 norm, also known as the Lasso, aims to shrink the estimated coefficients of the loss function. For an example linear regression:

$$Y \approx \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \qquad (8)$$

fitted using the residual sum of squares (RSS) loss function:

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \qquad (9)$$

we are able to shrink our coefficients by minimizing the RSS using a tuning parameter λ such that:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j| \qquad (10)$$

The L2 norm (also referred to as Ridge regression) also shrinks the coefficients of the model using λ, however we penalize all coefficients (β_j^2) rather than only the high coefficients ($|\beta_j|$) as in the L1 norm:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2 \qquad (11)$$

In the case of the L2 norm, the least important coefficients will be shrunk closely to zero, but never equal to exactly zero. Thus, our final model will still include all predictors. However, the L1 norm penalizes some coefficients to exactly zero at large enough λ, effectively removing these predictors from the model. The tuning parameter λ must be carefully selected to balance bias and variance. [105], [111]

2.4.4 Hyperparameter tuning

To best select the value of λ and other algorithm hyperparameters, there must be a validation dataset containing observations not included in the training dataset. When optimizing hyperparameters on the training data, there is a risk of overfitting and subsequent poor generalization of the model to unseen data. To circumvent this issue, the training dataset is split into two subsets—one to learn the parameters and another validation set to estimate the test error after training to update the hyperparameters. After hyperparameter optimization is complete, the error on the test set can be determined.

In addition to hyperparameter tuning and regularization, it is also important to consider the effect of our feature space on model capacity (i.e., underfitting and overfitting the model). When there are not enough features, underfitting results. When we have more features than observations, overfitting is risked. The difficulty of identifying the optimal number of features while balancing the consequences of each extreme is often referred to as the curse of dimensionality. As the dimension of our data increases, the data becomes "sparser" and more difficult to analyze. [112] Additionally, the risk of introducing noise (misleading or irrelevant data) also increases. The amount of training data necessary must also increase to mitigate these issues, substantially increasing the calculation time complexity. The number of training data needed increases exponentially in response to the number of features added—this becomes increasingly computationally complex and loses real-world value.

2.4.5 Feature selection methods

As a solution, dimensionality reduction is needed. Other benefits of reducing the dimension of our data (e.g., feature selection and feature engineering strategies) include decreasing the time-complexity and required memory storage to train the algorithm. Feature selection involves the identification of relevant features to be selected for model training. Feature engineering refers generally to the process of extracting useful features from raw data. [28], [112]

Commonly used feature selection strategies include those based upon variance threshold, univariate feature selection (e.g., Pearson correlation, ANOVA testing, or Chi-square algorithms in which the best *k* features, or a top percentile of features, are selected based off the information provided by these statistical measures), principal component analysis (PCA), linear discriminant analysis (LDA), multi-dimensional scaling (MDS), locally linear embedding (LLE), and t-distributed stochastic neighbor embedding (t-SNE). PCA and LDA are both based on linear transformations of data, where PCA focuses on features with maximum variance and LDA aims to maximize class separability. In contrast, MDS, LLE, and t-SNE are examples of non-linear manifold strategies that are capable of projecting complex data shapes into lower dimensional space while best preserving shape and Euclidean distance of data points. The mathematical basis and strategy for these dimensionality reduction techniques varies highly depending on the original data format and algorithm choice. [113], [114]

For non-structure-based feature selection methods in the machine learning models used for this project, the primary techniques presented for feature selection are recursive feature elimination (RFE) and partial least squares (PLS; including orthogonal partial least squares; O-PLS). RFE refers to a wrapper style algorithm in which features are ranked and selected by importance, where importance is determined by machine learning algorithm at the core of the model. [115], [116] This machine learning algorithm, such as logistic regression or random forest, is "wrapped" by RFE and features are ranked by a measure of feature importance (e.g., the coefficients of logistic regression or impurity-based node probability in random forest). The least important features are removed from the model, the model is re-fit, and this process is repeated until the user-specific number of features remains.

PLS aims to extract features from both the input matrix $\boldsymbol{X}$ and output matrix $\boldsymbol{Y}$ such that the covariance between all extracted factors is maximized. PLS is advantageous over PCA in its ability to address multi-colinearity (i.e., two or more data features are correlated with one another). [117]–[121] In comparison to PCA, which only considers the matrix of input variables without the context of the output data matrix, PLS considers the relationship between the input and output. The goal of PLS is to find a linear decomposition of $\boldsymbol{X}$ and $\boldsymbol{Y}$ such that:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E} \qquad (12)$$

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{Q}^T + \boldsymbol{F} \qquad (13)$$

where $\boldsymbol{T}$ represents the $\boldsymbol{X}$-factors, $\boldsymbol{U}$ represents the $\boldsymbol{Y}$-factors, $\boldsymbol{P}$ represents the $\boldsymbol{X}$-loadings (i.e., the linear coefficients that link terms to the factors), $\boldsymbol{Q}$ represents the $\boldsymbol{Y}$-

loadings, $\boldsymbol{E}$ represents the $\boldsymbol{X}$-residuals, and $\boldsymbol{F}$ represents the $\boldsymbol{Y}$-residuals. In factor analysis techniques such as PCA and PLS, loadings refer to the correlation coefficients between the variable and factor. Factor loading shows the variance explained by a variable on its given factor. The residuals, or error terms, are used to evaluate the accuracy of the factor analysis. The linear decomposition of $\boldsymbol{X}$ and $\boldsymbol{Y}$ is computed to maximize the covariance between $\boldsymbol{T}$ and $\boldsymbol{U}$; by extracting the factors that successively achieve this task, the best explanation of the $\boldsymbol{X}$-space, $\boldsymbol{Y}$-space, and the greatest correlation between $\boldsymbol{X}$ and $\boldsymbol{Y}$ are also maximized.

Each extracted X-score is a linear combination of $\boldsymbol{X}$; i.e., the 1st $\boldsymbol{X}$-score t of $\boldsymbol{X}$ is equal to $\boldsymbol{X}w$, where w is the eigenvector corresponding to the 1st eigenvalue of $\boldsymbol{X^T Y Y^T X}$. The same is true for the 1st Y-score $u = \boldsymbol{Y}c$ where c is the eigenvector corresponding to the 1st eigenvalue of $\boldsymbol{Y^T X X^T Y}$. After factors are extracted for each ith iteration, X and Y are deflated to equal $\boldsymbol{X}_i = \boldsymbol{X} - tt^{\mathrm{T}}\boldsymbol{X}$ and $\boldsymbol{Y}_i = \boldsymbol{Y} - uu^{\mathrm{T}}\boldsymbol{Y}$. This process is repeated until all possible latent factors t and u are extracted, i.e., when X is reduced to a null matrix. [117], [118], [122], [123]

O-PLS is a method based on the PLS algorithm that uses orthogonal signal correction to maximize the explained covariance for the 1st latent factors while the future factors describe variance in the predictors that is orthogonal to the response variables, i.e., uncorrelated to the response. O-PLS functionally differs from PLS in that the O-PLS algorithm filters out noise in the data by separately modeling variations of the $\boldsymbol{X}$-factor features that are correlated *and* uncorrelated to the $\boldsymbol{Y}$-factor features, respectively. This

key difference ultimately reduces model complexity, lowers the number of extracted factors, and allows for analysis of the source of orthogonal variation. [121], [124]

The O-PLS linear decomposition is mathematically similar to PLS, but differs in the inclusion of the orthogonal $\boldsymbol{X}$-factors and $\boldsymbol{X}$-loadings via a $\boldsymbol{T}_{orth}\boldsymbol{P}_{orth}^{T}$ term such that:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{T} + \boldsymbol{T}_{orth}\boldsymbol{P}_{orth}^{T} + \boldsymbol{E} \qquad (14)$$

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{C}^{T} + \boldsymbol{F} \qquad (15)$$

It is of note that while neural networks can be improved upon by preprocessing with a feature selection algorithm when appropriate, this is not always necessary as neural networks are capable as functioning as feature selection algorithms in themselves. Neural networks may identify salient features via backpropagation, updating the weights of the neural network. The mechanism of backpropagation and structure of neural networks is further described in further introductory *Neural Network* sections. [99], [102]

2.4.6 Machine learning algorithms

Many commonly used machine learning algorithms are implemented in the research plan proposed as predictive classifiers including regression, decision trees, AdaBoost, support vector machines (SVMs), and neural networks. [102], [125]–[132]

2.4.6.1 Regression

. Linear regression is a common supervised machine learning algorithm with a constant slope and a continuous output variable. Linear regression models are often presented in the slope-intercept form:

$$y = mx + b \qquad (16)$$

where y is the output variable determined by a variable (or feature) x multiplied by our slope coefficient m (otherwise referred to as the weight associated with each feature) and b is the y-intercept (otherwise referred to as the bias) that offsets our prediction. The goal of the algorithm is to predict the correct values for m and b to approximate the line of best fit. (**Figure 7**) [102], [105]

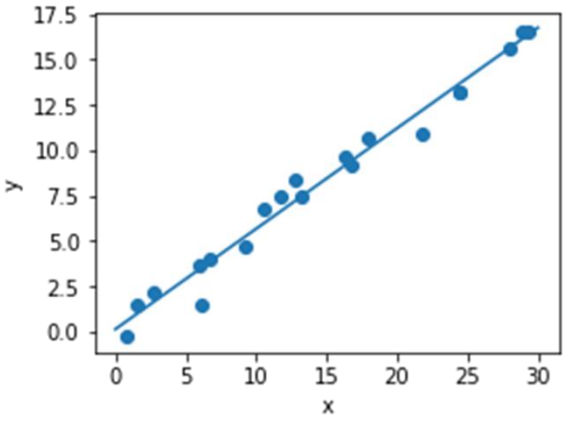


Figure 7. Linear regression model. An example of a linear regression line of best fit estimating the relationship between an input variable *x* and output variable *y* for a random dataset.

To optimize the weights of the model, a cost function is used. This process is generally applicable to any machine learning algorithm, but linear regression will be used as an example. Consider the MSE cost function applied to a linear regression model:

$$MSE = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2 \qquad (17)$$

where $y_i = mx_i + b$ for each i^{th} true value of x and $\hat{y}_i = mx_i + b$ is the i^{th} prediction for each value of y_i. MSE measures the average squared difference between the actual and predicted output values for each input value. The MSE cost changes with the current set of weight and should be minimized to improve the accuracy of the model. [102], [103]

Gradient descent is used to calculate the gradient of the cost function. The error associated with the weight is minimized via using the derivative of the cost function to compute the gradient (i.e., the slope of the cost function) and error is subsequently adjusted to move in the opposite direction of the gradient. The gradient can be split into the partial derivatives of the function of x and the function of m and b:

$$\big(y_i - (mx_i + b)\big)^2 = A\big(B(m,b)\big) \qquad (18)$$

$$A(x) = x^2 \qquad (19)$$

$$\frac{df}{dx} = A'(x) = 2x \qquad (20)$$

and

$$B(m,b) = y_i - (mx_i + b) = y_i - mx_i - b \qquad (21)$$

$$\frac{dx}{dm} = B'(m) = 0 - x_i - 0 = -x_i \qquad (22)$$

$$\frac{dx}{db} = B'(b) = 0 - 0 - 1 = -1 \qquad (23)$$

Combining these parts via the Chain rule yields:

$$\frac{df}{db} = \frac{df}{dx}\frac{dx}{db} \qquad (24)$$

$$\frac{df}{dm} = A'\big(B(m,f)\big)B'(m) = 2\big(y_i - (mx_i + b)\big) \cdot -x_i \qquad (25)$$

$$\frac{df}{db} = A'\big(B(m,f)\big)B'(b) = 2\big(y_i - (mx_i + b)\big) \cdot -1 \qquad (26)$$

Thus the gradient of the cost function is computed as:

$$f'(m,b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum -x_i \cdot 2(y_i - (mx_i + b)) \\ \frac{1}{N}\sum -1 \cdot 2(y_i - (mx_i + b)) \end{bmatrix} \qquad (27)$$

$$= \begin{bmatrix} \frac{1}{N}\sum -2x_i(y_i - (mx_i + b)) \\ \frac{1}{N}\sum -2(y_i - (mx_i + b)) \end{bmatrix} \qquad (28)$$

The model is trained via iterating through the dataset multiple times to improve the prediction via updating the weight and bias values in the direction indicated by the gradient of the cost function. [133], [134]

Training is considered complete based on an acceptable error threshold, the cost function converging (i.e., the algorithm ceases to further reduce cost with further iterations), or a set number of iterations. In addition to hyperparameters such as number of iterations, we also must choose a learning rate. The learning rate refers to the step size used in gradient descent. A greater value will learn faster but risks a suboptimal final prediction, while a smaller learning rate takes longer to converge. A learning rate that is too small risks the cost function never converging. [102], [105]

Lastly, the trained model is evaluated on a test set of data to evaluate its generalization error and predictive ability via measures of accuracy. This general process of tuning hyperparameters, training the model to update weights and reduce loss, and testing the model on a previously unseen dataset can be generally applied to other machine learning algorithms with different structures as well. [102]

In a multivariate regression (i.e., multiple input variables or features), the model resembles:

$$f(x_1, x_2, \ldots, x_n) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b \qquad (29)$$

Each input feature x_n has an associated weight that must be optimized in the learning process. As the number of features increases (i.e., our input dimensionality increases), the computational cost of calculating the gradient also increases. In addition to dimensionality reduction strategies, rescaling the data also aids in reducing time complexity. Common rescaling techniques include normalization (i.e., rescaling values to a range of [0,1]) and standardization (i.e., rescaling values to have a mean of 0 and standard deviation of 1). [105]

Linear regression models are only appropriate when the output variable is continuous. In contrast, logistic regression models are used when the output variable is categorical. Logistic regression models have the form:

$$\log(\eta) = \frac{1}{1 + exp\ (-\eta)} \qquad (30)$$

Logistic regression models output values between 0 and 1. (**Figure 8**)

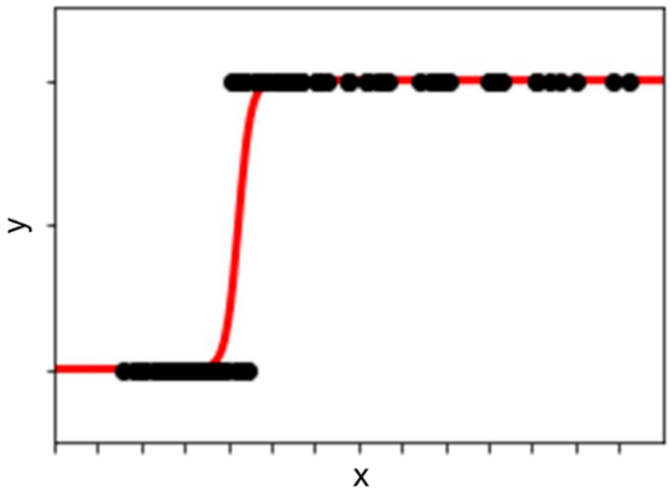


Figure 8. Logistic regression model. A logistic regression model has a characteristic sigmoidal 'S' shape predicting output *y* values between 0 and 1 from an input value *x*.

The logistic regression model finds a decision boundary, or threshold, to determine what class prediction each value will belong to. For example, in a binary classification with outputs [0,1], a threshold value of 0.5 corresponds to output values below 0.5 belonging to the ‘0’ class and otherwise belonging to the ‘1’ class. Linear and logistic regression models have many real-world similarities in that while they are both widely used for appropriate applications, they are often not appropriate for handling more complex data sets with higher dimensionality. [105]

2.4.6.2 Decision tree-based models.

Random forest is a supervised learning model that has gained popularity for its applicability and interpretability. Random forest can be used for prediction of both continuous and categorical output variables. In random forest models, the "forest" ensemble of decision trees built are trained via bagging to increase overall prediction accuracy. (**Figure 9**) [125], [126]

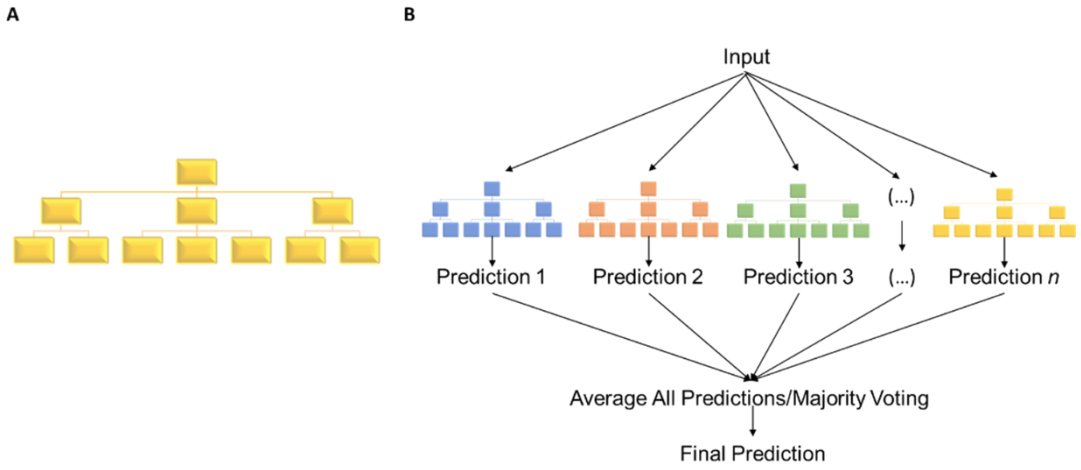


Figure 9. Decision tree and random forest models. A Schematic example of a simple decision tree. Starting from Row 1, there is one decision to be made with three possible outcomes. Outcomes #1 and #3 each have two resulting potential outcomes, while outcome #2 has three potential outcomes. **B** Schematic example of a simple random forest model in which the final prediction is generated via bagging of multiple decision trees.

Decision trees are a decision analysis tool that uses a branched flowchart model of decisions and their possible outcomes with associated probabilities. [105], [135], [136] The structure of a decision tree results in straightforward measurability and interpretability for the relative feature importance of each feature on the prediction. In a decision tree, each node represents the possible values of each attribute, each branch

represents the outcome of the decision, and each leaf node represents a class label (i.e., the final decision output after considering all features). A node that has no children is referred to as a leaf. Feature importance can be calculated via analyzing the extent to which each tree node for a given feature reduces impurity (the probability of incorrectly classifying a randomly chosen observation given the class distribution of the dataset) across all trees. Feature importances are scaled so the total sum of all importance is equal to 1. [102], [105] Many other variations of decision tree classifiers similar to random forest exist in the literature. These similar algorithms include the scikit-learn meta estimators that fit a number of decision trees on various subsets of the original data and improves predictive accuracy via aggregating or averaging individual predictions to form a final prediction, such as the bagged trees classifier and extra trees classifier. [115]

Random forest models are advantageous compared to simple decision trees despite the increased computational cost because the algorithm randomly selects a subset of data and features to build *several* decision trees and averages the results, increasing overall prediction accuracy. Additionally, random forest is less prone to overfitting than simple decision tree models. [105], [123]

Random forests are controlled by hyperparameters such as number of estimators (the number of trees used), maximum number of features used, the minimum number of leaves required to split a node, and out-of-bag (oob) sampling in cross-validation. [137]

In contrast to the bagging method used in random forest, adaptive boosting (AdaBoost) combines the output of other ‘weak learner’ algorithms (often, but not always, decision trees) into a weighted sum that represents the final output of the boosted

classifier. [127], [128] The AdaBoost classifier selects only the features known to improve the predictive accuracy of the model. Boost classifiers can be represented via:

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \qquad (31)$$

where f_t is a weak learner that takes an input x and returns a class label. The boosted classifier is trained so that the sum training error E_t is minimized:

$$E_t = \sum_{i} E[F_{t-1}(x_i) + \alpha_t h(x_i) \qquad (31)$$

Here $F_{t-1}(x)$ represents the boosted classifier up until the previous training iteration, $E[F]$ is some error function, and $\alpha_t h(x_i)$ is a weak learner $f_t(x)$ that is being added to the final boosted classifier wherein at each iteration t, each $f_t(x)$ is assigned a coefficient α_t and produces an output hypothesis $h(x_i)$. (**Figure 10**) Training iterations are informed by weights $\{w_{i,t}\}$ equal to $E(F_{t-1}(x_i))$. [102]

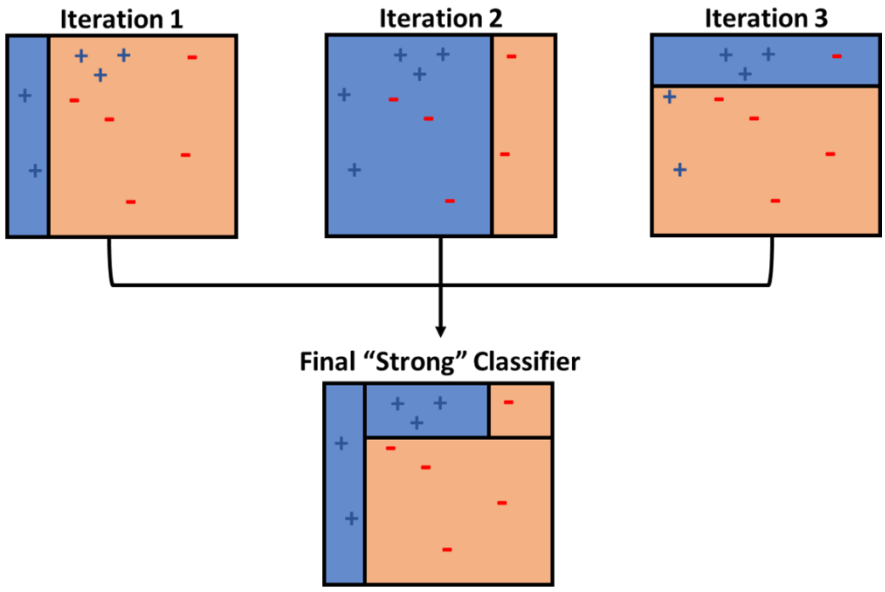


Figure 10. AdaBoost model. A data set of blue '+' symbols and red '-' symbols is classified by a cartoon representation of the AdaBoost algorithm. In each iteration, a "weak" learner attempts to separate the data. These "weak learners" are combined into a final weighted classifier to produce the strongest separation result.

2.4.6.3 Support vector machines.

SVMs are supervised learning models that are often applied to classification problems. SVMs aim to find a hyperplane or set of hyperplanes that best separate the data points into their respective classes. In geometry, a hyperplane refers to a subspace whose dimension is one less than that of its surrounding space (e.g., in a three-dimensional space, two-dimensional hyperplanes exist). The optimal hyperplane is known as the maximum-margin hyperplane, defined as the hyperplane that has the largest distance to the nearest data point of each class. (**Figure 11**)

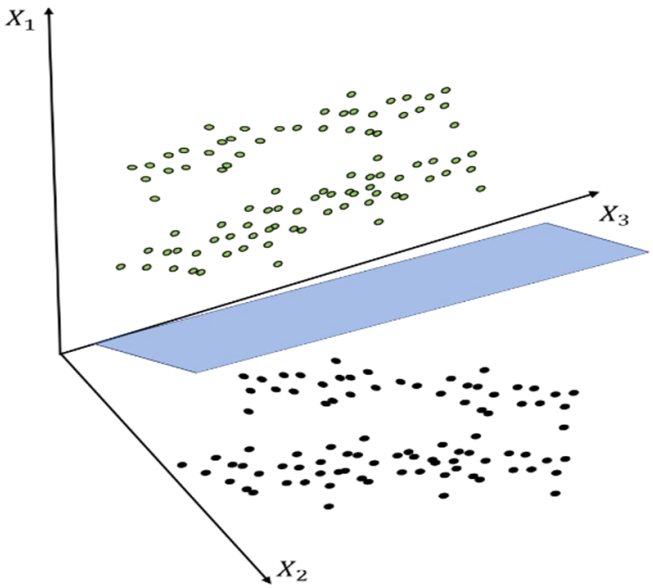


Figure 11. Support vector machine. A dataset is separated by a hyperplane. In a SVM, the maximum-margin hyperplane is used to maximize the distance between each class and the hyperplane, chosen based on the minimum test error possible. As an example, the hyperplane for 3-dimensional data is 2-dimensional.

The maximum-margin hyperplane is chosen as it typically results in the lowest test error for the classifier. Given a training dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where each x_i represents a vector of data features for a given i^{th} observation and y_i represents the corresponding discrete class label for said observation (e.g., 0 and 1 for a binary classification problem). A given hyperplane containing the set of points $\boldsymbol{x}$ can be written in the form:

$$\boldsymbol{w}^T\boldsymbol{x} - b = 0 \qquad (32)$$

where $\boldsymbol{w}$ is the normal vector to the hyperplane and b is the bias term. For a binary classification example, if the data are linearly separable, two hyperplanes are selected that separate the two classes of data such that the distance between said planes (i.e., the margin) is maximized. The maximum-margin hyperplane lies at the halfway point of the margin. The SVM is optimized via minimization of $\|\boldsymbol{w}\|$. In cases where the data are not linearly separable, techniques such as the hinge loss function can penalize for data points x_i on the incorrect side of the margin where the function equals:

$$\max(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i - b)) \qquad (33)$$

such that the function equals 0 when x_i is correctly classified while for incorrectly classified observations, the function's value is proportional to the distance from the margin. [129], [130]

Non-linear classifiers can also be constructed via application of the kernel trick, including common examples such as polynomial kernels:

$$k(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\overrightarrow{x_i} \cdot \overrightarrow{x_j})^d \qquad (34)$$

and the radial basis function [105], [130]:

$$k(\overrightarrow{x_i}, \overrightarrow{x_j}) = \exp\left(-\gamma \left\| \overrightarrow{x_i} - \overrightarrow{x_j} \right\|^2\right) \text{ for } \gamma > 0 \qquad (35)$$

2.4.7 Neural Networks

In 1958, Rosenblatt created the single-layer perceptron. [131] A perceptron, essentially a single node (also called a neuron) activated by the Heaviside step function, takes the form:

$$f(x) = \begin{cases} 1 \text{ if } \boldsymbol{w} \cdot \boldsymbol{x} + b > 0, \\ 0 \text{ otherwise} \end{cases} \qquad (36)$$

where the input $\boldsymbol{x}$ is mapped to an output value $f(x)$ and $\boldsymbol{w}$ is a vector of real-valued weights and b is the bias. While the single-layer perceptron is the simplest feedforward neural network, modern networks typically contain multiple functions. Feedforward neural networks, also known as multilayer perceptrons, are the simplest modern form of a neural network and the foundation upon which other neural networks with increasing complexity are created. The term 'feedforward' refers to the way that information flows or transfers through the model via the initial input $\boldsymbol{x}$ via a series of intermediate functions to produce the output $f(x)$. The function is referred to as a 'network' because the input undergoes a series of transformations via a chain of connected functions typically represented as a graph of nodes. (**Figure 12**) The term 'neural network' comes from the idea that neural networks are loosely modeled off of the structure of biological neurons, but this is considered to be a misnomer as the similarities are limited. [99], [102], [105], [138], [139]

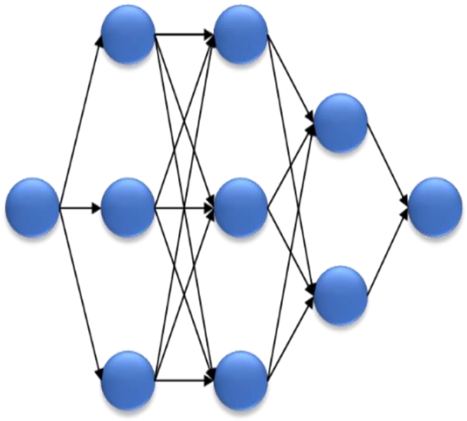


Figure 12. Neural network structure. A cartoon schematic showing the generalized structure of a deep neural network containing an input layer, three hidden layers, and one output layer. Data from the single input node is transformed by an activation function at each node in the following layer. Each connection is weighted, and these weights are adjusted as the network learns.

Each 'neuron' in the neural network is represented as a node in the network. The overall length of the chain of nodes gives the depth of the network. Generally, neural networks with more than two layers are referred to as deep neural networks. For example, we may have three neurons representing the functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ connected via a network to form $f(x) = f^{(3)}\left(f^{(2)}\left(f^{(1)}(\boldsymbol{x})\right)\right)$. The network is structured into layers containing nodes that act in parallel where each node represents a vector-to-scalar function. The dimensionality of each layer is called the width of the network. The layers between the input layer and output layer are referred to as hidden layers, named in reference to the 'black box' nature of neural networks. As data is passed through each layer and the neural network begins to approximate the form of $f(x)$, the output for each layer is not shown; rather, it is simply fed into the next layer until the final output layer. The nodes in each layer are connected and each connection is assigned a weight representing its relative importance to the model. The neurons within the hidden layers typically have multiple input and output connections. If every neuron in a layer connected

to each neuron in the following layer, the layers are fully-connected. In contrast, when a group of neurons in one layer are all condensed and connected into the same neuron in the following layer, it is called pooling. Recurrent networks, in contrast to feedforward networks, allow connections between neurons in the same layer or previous layers. [99], [102], [105], [138], [139]

The hyperparameters for a feedforward neural network include learning rate, depth, width, and batch size. More complex neural network structures, such as recurrent and convolutional neural networks, have their own additional unique set of hyperparameters. [140]

In a feedforward network with hidden layers, activation functions are used to compute the hidden layer values. In a neuron model consisting of an input $\boldsymbol{x}$ with bias b and a weight of $\boldsymbol{w}$ summarized together as z and an activation function φ that performs a non-linear transformation of z. The resulting output of the neuron model can be described as:

$$y = \varphi(z) = \varphi(\boldsymbol{w}^T\boldsymbol{x} + b) \qquad (37)$$

Many activation functions exist. Commonly found activation functions are summarized in **Table 1**. [102]

Activation Function	Hyperbolic tangent	Sigmoidal	ReLU	Softmax	Heaviside	Signum
φ	$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$S(x) = \frac{1}{1+e^{-x}}$	$R(x) = \begin{cases} 0\ for\ x < 0 \\ x\ for\ x \geq 0 \end{cases}$	$y_i = \frac{e^{x_j}}{\sum_j^n e^{x_j}}$	$H(x) = \begin{cases} 0\ for\ x < 0 \\ 1\ for\ x \geq 0 \end{cases}$	$\begin{cases} -1\ for\ x < 0 \\ 0\ for\ x = 0 \\ 1\ for\ x > 0 \end{cases}$
φ'	$1 - \varphi(x)^2$	$\varphi(x)(1 - \varphi(x))$	$\begin{cases} 0\ for\ x < 0 \\ 1\ for\ x \geq 0 \end{cases}$	$\frac{\delta y_i}{\delta j} = y_j(\delta_i - y_j)$	$\delta(x)$	$2\delta(x)$
Range	$(-1, 1)$	$(0, 1)$	$[0, \infty)$	$(0, 1)$	$[0, 1]$	$[-1, 1]$

Table 1. Common activation functions for neural networks, their first derivatives, and ranges.

The computation of gradients and updating of weights in neural networks is performed via an algorithm called backpropagation. Backpropagation calculates the gradient of the cost function associated with a given set of weights corresponding to each connection of the network. Backpropagation involves computing the gradient of the loss function via the chain rule one layer at a time, passing through the network backwards starting from the last layer to prevent unnecessarily repetitive calculations within the Chain rule. This method is much more efficient than a direct computation of the gradient with respect to each individual weight. Via this process, weights are iteratively updated with each epoch of training such that the weights of the least important neurons are decreased and the weights of the more important neurons are increased proportionately. [102], [141]–[143]

There are many specialized types of neural networks, including the convolutional neural network (CNN). CNN have been previously applied to biomedical problems with success due to their ability to analyze three-dimensional molecular structure data. [19], [21], [98], [139] CNN utilize convolutional and pooling layers with local connectivity between neurons (i.e., neurons are only connected to the nearby neurons in the next layer rather than fully connected). All of the connections between a group of locally connected neurons, called local receptive fields, use a set of weights denoted as a kernel. A kernel will be shared with all of the other neurons that connect to their local receptive fields and the results are stored in a matrix called the activation map. This process is described as weight sharing. Weight sharing and the local connectivity property allow a CNN to handle data with high dimensions. [99], [102], [144]–[147] Pooling layers aim to reduce the input dimension and aid in improving the generalization of the network. The pooling

layer scans the entire input in the same manner as a kernel in a convolutional layer. The most common type of pooling is max pooling, where the maximum value within each specified sub-window is extracted across the activation map. For example, a pooling layer with a stride of two and window size of two will half the size of the input dimension. [148]–[151] **Figure 13** represents the overall structure of a CNN.

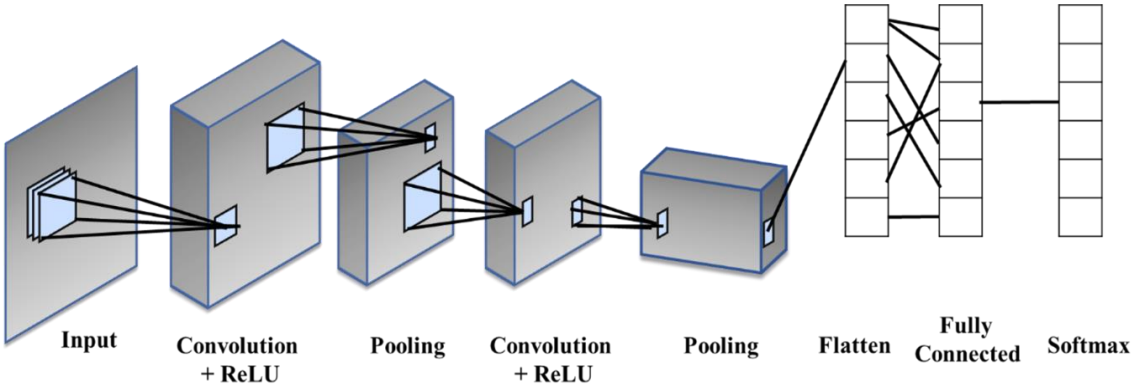


Figure 13. Convolutional neural network architecture. A cartoon schematic showing the generalized structure of a convolutional neural network containing pooling layers. Convolutional layers extract high-level features such as edges in an input image or gradient. Added convolutional layers contribute to a higher level of feature extraction. Pooling layers reduce the spatial size of the convolved features to decrease the associated computational cost necessary via extracting only the dominant features.

2.5 Bioinformatics methods.

2.5.1 Molecular dynamics simulation

Molecular dynamics simulation and molecular modeling technology have been key techniques used in the advancement of biochemical research; in particular, SBDD. Ligand docking is a core modeling protocol valued for its significant time- and cost-efficient advantage over traditional biochemical protocols. While computational advancements have improved the time complexity and accuracy of simulation techniques, high dimensionality and throughput capacity are still concerns when using modeling alone in bioinformatic studies. [94] Recent molecular dynamics studies have shown success at elucidating the structural details of specific molecular interactions, while large scale studies have been increasingly dependent on machine learning methods. [35], [92], [96], [152], [153]

Within recent years, machine learning tools capable of handling “big data” problems have been applied to computational biochemistry research to address the large feature space and amount of output data produced when studying molecular interactions. [19]–[22], [98], [100], [138], [154], [155] Augmenting biomedical research with machine learning has become increasingly common over the last thirty years as high throughput techniques advance and produce large amounts of data. Machine learning is often applied to the fields of systems biology, genomics, proteomics, pharmacology/toxicology, and structural biochemistry. [138] The field of computational toxicology often uses

quantitative structure-activity relationship (QSAR) modeling to predict the activity of a compound against a set of biological targets. [156] QSAR programs often use regression models to predict the biological activity of a chemical against protein receptors trained on experimental assay data. Many QSAR models use ToxCast data as training sets and focus on respiratory, skin, hepatic, renal, or neural toxicity. [157]–[161] Within the realm of structural biochemistry, machine learning programs have proven useful for protein structure prediction and classification as well as interpretation of molecular dynamics output data. [98], [155], [162]

2.5.2 Machine learning models for bioinformatics

Machine learning programs for molecular applications use chemical descriptors as features to describe the data and aid in the identification of a pattern within the data. [163] These descriptors may be macromolecule specific (e.g., amino acid descriptors for proteins) or generalized physical chemistry descriptors such as electron density, nucleophilicity, charge, or topological indices. Molecular topology describes the geometry and structure of a chemical given the atomic positions and chemical bonding behavior of a given molecule. Topological indices are very useful features in machine learning applications where the goal is to predict biological activity based on the chemical structure of a molecule. [164], [165]

Many algorithms have been trained specifically for structural analysis of protein structure and function, including the recent SSnet [22], DeepScreen [20], DeepAtom [19], and DeepConv [21] methods described in *Specific Aim 1*. Despite the established progress of augmenting protein biochemistry research with machine learning, current models often produce suboptimal accuracies for reliable clinical translation and few examples of transfer learning to a specific clinical application exist. [139] GPCR-focused models primarily include QSAR analysis, class prediction, and evolutionary biology studies but the field predominately still relies on computationally-expensive molecular dynamics methodology. [29], [84], [166]–[168] Within the previous two years, a limited number of GPCR-ligand interaction studies utilizing machine learning have been

published, demonstrating the potential success for a similar model to be developed specifically against a given GPCR subtype. [169], [170]

Classification and behavior prediction of protein receptors may use feature descriptors based on sequence, structure, or both. Tens of thousands of protein chemical descriptors exist, and a complete comparison of all available feature groups and strategies is beyond the scope of this work. [163], [171]–[177] Protein sequence descriptors may include polarity, charge, and size of amino acids. In contrast, protein structural features take into account the three-dimensional shape of a protein and the intramolecular interactions between amino acids as well as the protein interactions with solvent. [164], [178] Protein structural features may offer more useful information than sequence alone as structure is an important determinant of function. [179]–[181]

2.5.3 Anisotropic Network Model

Atomic interaction data from normal mode analysis models such as the Gaussian Network Model (GNM) and the Anisotropic Network Model (ANM) provide detailed structural information that is directly related to the molecular dynamics and function of a protein comparable to molecular dynamics simulation at a fraction of the computational cost. [182]–[191] These elastic network modeling tools are specifically tuned to examine proteins via the relationship of force constants versus the interparticle distance between the residue alpha carbons (C_α). In these models, the protein is represented as an elastic mass-and-spring network where each network node is a residue's C_α and each $C_\alpha - C_\alpha$ interaction is represented as a spring, based on the concept that the atomic fluctuations in a biomolecule are subject to harmonic potential much like masses connected to springs, pendulums, or acoustics. The harmonic potential between all interacting C_α nodes describes the internal motions of the "spring" connecting each $C_\alpha - C_\alpha$ pair. The ANM is an augmentation of the GNM as the GNM does not account for directionality of fluctuation while the ANM uses three-dimensional atomic coordinates to account for directionality when describing the compression and expansion of the spring. [184], [186] In the GNM, there is no directional preference or three-dimensional character of motion accounted for, therefore, the molecule is views as a collection of N nodes (i.e., one for each residue). The resulting ensemble has $N - 1$ independent modes, where a "normal mode" or harmonic refers to the pattern of motion in an oscillating system in which all

system components move sinusoidally at the same frequency. In the three-dimensional system of an ANM, $3N - 6$ modes are obtained. It has been demonstrated in the literature that atomic fluctuations are generally anisotropic, or capable of changing properties in different directions (e.g., properties such as absorbance, conductivity, and polarizability are anisotropic in nature). These directional motions are directly related to the biological functions of proteins. Cumulatively, this evidence supports the notion that the meticulously detailed nature of the information provided by the ANM balanced with the relatively simple time complexity of calculations provided by the lack of necessity to differentiate between the chemical differences of specific amino acids makes feature generation and feature engineering via the ANM an advantageous approach for protein structure classification models. [184], [185], [192]

The ANM is useful in describing the atomic fluctuations and molecular dynamics of proteins to better describe their three-dimensional structural and functional characteristics. This information can be used to guide molecular dynamics simulations as well as provide a rich source of features to be used in training machine learning models. [185] Normal mode analysis methods like the ANM are advantageous over full scale molecular dynamics simulation because they are capable of representing large scale atomic behavior at a fraction of the computational cost with robust results. [164], [193], [194] Using features provided by ANM analysis has shown promise as a comparable method for describing protein structure in machine learning models to the well-established RMSD and TM-scoring strategies. [192] In general, the ANM is more accurate for globular proteins than non-globular, proteins with high-resolution structures, buried residues as compared to surface residues, and polar residues as compared to

hydrophobic residues. [185] The ANM has shown success in the analysis of the structural characteristics of many proteins including hemoglobin [195], amylase [184], HIV-1 reverse transcriptase [182], [196], DNA polymerase [197], CoV spike protein [198], and numerous GPCRs [199], [200].

2.5.4 Protein structural comparison methods

Protein structure can be described independently or in comparison to other protein structures. By comparing individual structures to all others in a dataset, it is possible to quantify structural variation and identify the most unique regions of the structural landscape. [192] Variation can be determined via measures such as root-mean-square deviation of atomic positions (RMSD), TM-score, and variance threshold-based feature importance ranking of chemical descriptors like those obtained from normal mode analysis. [192], [201]

RMSD is a measure of the distance between pairwise atomic coordinates between two superimposed protein structures. RMSD is defined by:

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2} \qquad (38)$$

where the averaging is performed over the n pairs of aligned atoms and d_i is the distance between the two atoms in each i^{th} pair, presented in Å units. [201]

RMSD is calculated between the alpha carbon (C_α) of paired equivalent atoms for any subset (or all) residues between two proteins.

The key disadvantage when using RMSD is the magnitude by which it is swayed by the amplitude of error. Two structures that are highly similar or identical with each other

with the exception of a single region of difference (such as a disordered loop or terminus) will have a very large RMSD and cannot accurately be superimposed. A pair of proteins that only differ in one region will have a similar global backbone RMSD to that of a protein pair with multiple small-scale differences throughout the entire structure. As s a result, RMSD scores are strongly affected by both flexible and poorly defined regions. [201]

RMSD scoring requires unambiguous establishment of atom pair correspondence which is problematic when considering residues that possess internal symmetry or multiple equivalent side chain rotamers. Typically, assignment of atom pair correspondence is established trivially via sequence alignment. However, for Arg, Asp, Glu, Leu, Phe, Tyr, and Val residues this is not possible because some atoms are topologically equivalent due to sidechain internal symmetry. For example, the $C_{\delta 1}$ and $C_{\delta 2}$ atoms within a Phe residue are topologically equivalent and can thus be mapped into $C_{\delta 1}$ and $C_{\delta 2}$ atoms of the corresponding Phe residue in a paired structure in two different possible combinations. As a result, enumeration for the optimal rotamer of these residues is necessary when calculating RMSD. Finding the optimal rotamers can change pocket RMSD by up to 0.5 Å but this additional step increases the algorithm time complexity. [201]

Additionally, RMSD scoring is heavily dependent on protein length and size. TM-score measurements address this issue by normalizing the distances between corresponding atom pairs via applying a distance scale. In the TM-score calculation, a

target protein L_{target} and an aligned protein $L_{aligned}$ are assigned a score based off the normalized distance between the atoms such that:

$$TMscore = \max\left[\frac{1}{L_{target}}\sum_{i=1}^{L_{aligned}}\frac{1}{1+\left(\frac{D_i}{D_0\left(L_{target}\right)}\right)^2}\right] \quad (39)$$

RMSD and TM-scoring are the most common measures of structural similarity used when constructing protein structural alignments. In contrast to sequence-based alignment methods, three-dimensional structural alignments focus solely on the relative shape and positional similarity of protein structures based on atomic coordinates, irrelevant of the two-dimensional protein sequence. Closely aligned protein models indicate potential functional relationships that may not be apparent from sequence comparison alone. Structural alignment algorithms often agree with sequence alignments; however, many protein families have high structural and topological similarity despite low sequence overlap as protein structure is more robust to evolutionary changes as compared to sequence. [201], [202] As an example, the ubiquitous TIM barrel motif is found in over 70 different protein families despite sequence variations. While sequence-based techniques do not detect this similar motif accurately, structural alignments find meaningful patterns allowing for recognition of the ubiquitous TIM barrel. [180], [192]

In comparison to the use of traditional multiple structural alignment metrics like RMSD and TM-scoring, there is also evidence for success in the literature for using machine learning-based feature selection of structural features. The literature demonstrates the utility of GNM and ANM output features as determinants of

"informative residue positions." Following the generation of a multiple structure alignment and feature generation of GNM and ANM fluctuations and residue depths, features are aligned after discarding gap positions in the alignment. The data was fit using a logistic regression model with Lasso regularization and the absolute value of the feature coefficients at each position was used to determine feature importance. This method resulted in an AUC-score of 0.98 ± 0.03 for a cyclin-dependent kinase classification model. [192] This method serves as strong evidence for the applicability of physical chemistry modeling and structural analysis to machine learning feature engineering.

3. Methodology

3.1 Literature search and model selection

A literature search was conducted for journal articles in the NCBI PubMed database within the previous five years containing any or all of the following terms: “neural network”, “deep learning”, “machine learning”, “prediction” + “protein binding”, “receptor”, “ligand binding”, “drug binding”, “protein-ligand interaction”, “drug-target interaction”. Results were filtered based on classifier output to obtain published architectures where the output predicts interaction activity between a protein and a ligand binding partner (i.e., as compared to prediction of binding site location, free energy, et cetera). Out of 293 total results, four papers were selected for comparison based on superior performance metrics. The selected papers describe architectures referred to as DeepConv, SSNet, DeepScreen, and DeepAtom, respectively. [19]–[22] While each model is reported to have high accuracies, weaknesses in their general applicability as indicated by strict data input requirements, large dataset necessity, and inconsistent results against varying test datasets still demonstrate a gap in the literature for improvement. **Table 2** summarizes the architecture, requirements, strengths, and weaknesses of each model.

Model	DeepConv	SSNet	DeepScreen	DeepAtom
Year Published	2019	2021	2020	2019
Reported Performance	AUC = 0.852	AUC = 0.98	F1 = 0.87	R = 0.83
Implementation	Keras with tensorflow	Keras with tensorflow	PyTorch	PyTorch
Model Architecture Summary	Sequence data → Embedding Layer → Convolution along the sequence → Global Max-Pooling Layer → Output Score	Backbone curvature → four convolutional neural networks (32:64:128 filters into two Global Max-Pooling Layers each) → protein concatenate layer -> protein ligand concatenate layer → Output Score	2D Ligand Images → Target Protein-Specific Convolutional Network (Convolutional Layer → Pooling Layer → Convolutional Layer → Pooling Layer) → Output Classification	Structure voxelization → Feature Extraction Block (three 3D shuffle group layers) → 3D Max Pooling Layer → 3D Pointwise Conv → 3D Protein-Ligand Complex
Strengths	Large training dataset with wide breadth of protein families	Applicable to a wide range of molecular complexes	Large training set; well-balanced classes	Accounts for three-dimensional structure with minimal feature engineering necessary
Weaknesses	Negative interactions generated by scrambling (i.e., not experimentally validated)	Small training set; only allows extreme values for dissociation constant (K_d<100nM or K_d>10000nM)	Requires complex structures	Computationally expensive (e.g., voxelized input and output); requires complex structures

Table 2. Comparison of Selected Protein-Ligand Interaction Deep Learning Models. DeepConv, SSNet, DeepScreen, and DeepAtom models are summarized for its implementation strategy, performance and model strengths, and model weaknesses.

3.2 Data collection and cleaning

Benchmark test data for Specific Aim One were adapted from the Database of Useful Decoys – Enhanced (DUD-E). [25] Each selected model requires different input formats, including ChEMBL entries, sequence data, apoprotein structure, and protein-ligand complex data. The original complete DUD-E dataset was filtered for interactions that contain all of these available data, resulting in 438 interactions spanning 11 protein receptors. This dataset was relatively class balanced with approximately 56% of data observations belonging to the positive class. In addition to the benchmark test set, a small test set of Subgroup A17 (serotonin and dopamine) receptors was generated containing 37 interactions spanning 4 protein receptors.

To correct a strong class imbalance in the Subgroup A17 data (70.2% of data observations belonging to the positive class), the negative class was oversampled using SMOTE (see **3.3**) resulting in an equally balanced dataset. [203] The newly generated dataset with added synthetic datapoints contains 52 observations split between 26 positive and 26 negative interactions.

Specific Aim Two and Specific Aim Three datasets were adapted from the Protein Databank (PDB) [204], G Protein-Coupled Receptor Database (GPCRdb) [205][206], Psychoactive Drug Screening Program (PDSP) [26], and DrugBank [207] to obtain thorough, experimentally validated interaction data for Subgroup A17 positive and negative interaction behavior wherein ‘positive interactions’ refer to a drug-target pair

associated with a measured K_d / K_i < 10,000nM and 'negative interactions' refer to a drug-target pair associated with a measured K_d / K_i ≥ 10,000nM per previously published protocols in the literature.

For Specific Aim Two, a small dataset of Group A GPCR protein structures was collected to test the effect of structure-based feature selection methods on accuracy and time complexity of classification methods. Initially, 35 PDB structures of Group A proteins were collected based on exclusion criteria selecting for high resolution, high percent sequence coverage, active conformation apostructures. These 35 structures spanned ten different Group A subgroups, six of which contained less than two structures per subgroup. Proteins belonging to these smaller groups were removed to ensure quality classification input. The remaining subset of proteins included 29 proteins spanning six subgroups (A4, A12, A13, A15, A16, A18). Each protein is associated with a data tensor containing structural data from the PDB file including .xyz coordinate data, secondary structure information, and bond angles. This structural data was used to generate an anisotropic network model for each protein with a cutoff of 15A for alpha carbon interactions calculating 20 normal modes per protein structure (*see 3.4*). Eigendecomposition was computed using MATLAB. [208]

Alpha carbon fluctuations from each ANM model respectively were aligned using a multiple sequence alignment from the Caretta algorithm. [192] In total, 298 residues were aligned and following the removal of any positions with gaps, 174 positions corresponding to 174 total features remained comprising the fully comprehensive feature set. The dataset was imbalanced with a majority of the sample belonging to subgroup A18 therefore SMOTE was applied to the dataset. The final dataset contained 78

observations. Additionally, a limited feature set for these observations was subsetted based on the 25 residue positions with the largest magnitude variance values (15% of the original feature set) to test the hypothesis that structure-based feature selection methods are capable of reducing model training time without sacrificing model accuracy.

To determine the potential for learning pharmacodynamic behavior patterns from structural GPCR data, two datasets were generated containing pairs of either serotonin (5-HT) or dopamine (DRD) receptors respectively and corresponding ligands labeled for each receptor as an interacting pair (also referred to as a 'positive' or 'active' interaction or coded with an output of '1') or a non-interacting pair (also referred to as a 'negative' or 'inactive' interaction or coded with an output of '0'). Data was obtained from the GPCRdb, PDSP databases, and the DrugBank for GPCR structure, interaction data and psychoactive drug activities. [26], [205]–[207]

5-HT receptor data was joined from the three online databases using SQL to generate a single united dataset. Each database contains experimental values for the 5-HT1a, 5-HT1b, 5-HT2a, 5-HT2b, and 5-HT2c receptor isoform drug-target interactions. Additionally, PDSP and DrugBank contain additional experimental values for confirmed non-interacting ligands to obtain validated negative interaction examples. SQL join functions were used to filter each database for serotonin assay results, simultaneously removing irrelevant proteins and redundant data entries. A number of drug-target pair assay results were found to be duplicates with contradictory reported results, which is to be expected with differing conditions and assay kits within experimental research.

To address this dilemma objectively, the decision to include a drug-target pair was based on the agreement of results in reference to the 10,000nM decision boundary to discern between positive/negative interactions to reflect the boundary given in previously published models. [19-22] If each duplicate assay entry lies on the same side of the boundary, results are combined into a single observation based on the resulting binary output encoding. On the contrary, if duplicate assay results span a range included, but not bound by, 10,000nM, these results are contradictory and all results for the given drug-target pair are thrown out (e.g., if a given 5-HT receptor has three listed assay results with one result under 10,000nM and two results above 10,000nM, these results are contradictory; however, if one result equals 10,000nM and two results are above 10,000, these results are considered to be in agreement and will be combined into a single data observation encoded as '0'). In addition to addressing duplicate results, data was also cleaned for duplicate ligand entries under varying chemical names, null values, and incomplete entries. Although the 10,000nM boundary does not appropriately represent the biochemical significance of binding strength (e.g., the difference between a dissociation constant of 1nM and 900nM leads to a substantially different biochemical effect), it was not possible to accurately and objectively predict a continuous output variable of dissociation constants due to duplicate and contradictory results in the literature for certain binding interactions as described above. Clinically relevant examples of GPCR binding interactions run the gamut of binding affinities and multiple arguments could be made for various threshold boundaries to be of pharmacological importance. The scale of binding affinities of drugs is often much lower than the binding affinities of native ligands and could arguably deserve different threshold values. We should also

ideally consider factors such as transient binders, allosteric binders, and receptor mutations.

Because of our limitations with the number of variables to account for, training data required, maintenance of data quality, and factors involved in binding threshold determination, we chose to opt for a binary classification and use the same decision boundary as other models. Using the same decision threshold is also valuable when comparing model accuracies. Other ways that this could be handled other than the added complexity described above include a multiclass model where we have more decision boundaries to more specifically categorize binding interaction strength. The final clean 5-HT dataset contains 36,208 interactions with approximately 57% negative and 43% positive interactions.

Each protein is associated with a data tensor containing structural data from the PDB file including .xyz coordinate data, secondary structure information, and bond angles. This structural information was used to generate ANM for each 5-HT model as previously described for Specific Aim Two dataset creation methodology.

In total, 342 residues were aligned and following the removal of any positions with gaps, 282 positions corresponding to 282 total features remained comprising the fully comprehensive feature set. SMOTE was applied to the positive class to fix imbalance as this adjustment improved overall model accuracy. Additionally, ligand binding partners associated with a SMILES identifier for chemical structure were described by features extracted by the RDKit feature engineering program for a total of 201 features describing physiochemical, quantum, and topological descriptors of each molecule. ([209],

Appendix A) The final 5-HT dataset contained 42,656 observations consisting of two-dimensional vectors encoding 483 features describing protein and ligand structure. Additionally, a limited feature set for these observations was subsetted based on the 42 residue positions with the largest magnitude variance values (15% of the original feature set) based on evidence supporting the hypothesis from Specific Aim Two.

DRD data was collected, joined, and cleaned using an identical protocol to that of the 5-HT dataset. As a result, a dataset containing drug-target interactions classified by a binary positive/negative output value was generated for the DRD1, DRD2, DRD3, and DRD4 isoforms. DRD5 was excluded due to the lack of a high-resolution solved structure. Structural feature tensors were generated from PDB files as input for the ANM, and feature engineering from ANM results was conducted as previously described for the 5-HT isoforms. In total, 369 residues were aligned and following the removal of any positions with gaps, 302 positions corresponding to 302 total features remained comprising the fully comprehensive feature set. Synthetic Minority Oversampling Technique was applied to the positive class to fix imbalance as this adjustment improved overall model accuracy (*see 3.3)* . Ligand binding partners were described by 201 RDkit features as previously described. The final DRD dataset contained 2,063 observations consisting of two-dimensional vectors encoding 503 features describing protein and ligand structure. Additionally, a limited feature set for these observations was subsetted based on the 75 residue positions with the largest magnitude variance values (15% of the original feature set) based on evidence supporting the hypothesis from Specific Aim Two. All programing methodology is further detailed in **Appendix B**.

3.3 Synthetic datapoint generation

For all datasets, Synthetic Minority Oversampling Technique (SMOTE) was used to generate synthetic datapoints oversampling the minority class to achieve a more balanced dataset. [203] Briefly, SMOTE selects random datapoints from the underrepresented class and determines the *n*-nearest neighbors of a given datapoint. New data points are generated by selecting a random point on the vector that connects the original datapoint and a selected n_i neighbor. This process is repeated until the dataset is equally distributed. SMOTE was implemented using the imbalanced-learn module. SMOTE is a well-established, robust method with approximately two decades of published success in the literature. [210]

3.4 Anisotropic network model

ANMs can predict anisotropic motions of atoms via computation on the force constant matrix (***H***) containing data describing the orientation of each interaction with respect to the global coordinates where:

$$\boldsymbol{H} = \begin{bmatrix} H_{ii} & H_{ij} \\ H_{ji} & H_{jj} \end{bmatrix} \qquad (40)$$

where i and j are nodes in the model representing two C_α in the protein and each element H_{ij} in the matrix represents the anisotropic information regarding the orientation of nodes i and j. [184], [185] The Hessian matrix ***H*** describing the force constant of the system is equivalent of the second partial derivative of the harmonic potential $V_{i,j}$ defined as:

$$V_{i,j} = \frac{\gamma}{2}\left(s_{i,j} - s_{i,j}^{o}\right)^2 \qquad (41)$$

where γ represents the unknown spring constant, $s_{i,j}$ is the instantaneous distance between i and j, and $s_{i,j}^{o}$ is the equilibrium distance between i and j. Therefore:

$$\frac{\partial^2 V_{i,j}}{\partial x_i^2} = \frac{\partial^2 V_{i,j}}{\partial x_j^2} = \frac{\gamma}{s_{i,j}^2}\left(x_j - x_i\right)^2 \qquad (42)$$

$$\frac{\partial^2 V_{i,j}}{\partial x_i \partial y_i} = \frac{-\gamma}{s_{i,j}^2}\left(x_j - x_i\right)\left(y_j - y_i\right) \qquad (43)$$

where the instantaneous positions of atoms i and j are defined by their respective coordinates (x_i, y_i, z_i) and (x_j, y_j, z_j). As a result, $\boldsymbol{H}$ can be expanded as:

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial^2 V_{i,j}}{\partial x_i \partial x_j} & \frac{\partial^2 V_{i,j}}{\partial x_i \partial y_j} & \frac{\partial^2 V_{i,j}}{\partial x_i \partial z_j} \\ \frac{\partial^2 V_{i,j}}{\partial y_i \partial x_j} & \frac{\partial^2 V_{i,j}}{\partial y_i \partial y_j} & \frac{\partial^2 V_{i,j}}{\partial y_i \partial z_j} \\ \frac{\partial^2 V_{i,j}}{\partial z_i \partial x_j} & \frac{\partial^2 V_{i,j}}{\partial z_i \partial y_j} & \frac{\partial^2 V_{i,j}}{\partial z_i \partial z_j} \end{bmatrix} \qquad (44)$$

$$= \frac{-\gamma}{s_{i,j}^2} \begin{bmatrix} (x_j - x_i)(x_j - x_i) & (x_j - x_i)(y_j - y_i) & (x_j - x_i)(z_j - z_i) \\ (y_j - y_i)(x_j - x_i) & (y_j - y_i)(y_j - y_i) & (y_j - y_i)(z_j - z_i) \\ (z_j - z_i)(x_j - x_i) & (z_j - z_i)(y_j - y_i) & (z_j - z_i)(z_j - z_i) \end{bmatrix} \qquad (45)$$

$$= \frac{-\gamma}{s_{i,j}^2} \begin{bmatrix} x_j - x_i \\ y_j - y_i \\ z_j - z_i \end{bmatrix} [y_j - y_i \quad y_j - y_i \quad z_j - z_i] \qquad (46)$$

While the matrix $\boldsymbol{H}$ is not invertible, a pseudoinverse can be obtained via $\boldsymbol{H} = \boldsymbol{U}\Lambda\boldsymbol{U}^T$ to obtain the eigenvectors and nonzero eigenvalues. The eigenvalues describe the vibrational direction and amplitude in the different modes. The mean square fluctuations of specific residues can be obtained by summing the fluctuations in each individual mode, i.e.,

$$\langle \boldsymbol{\Delta R_i^2} \rangle = \langle \Delta X_i^2 \rangle + \langle \Delta Y_i^2 \rangle + \langle \Delta Z_i^2 \rangle = \frac{k_B T}{\gamma} \left[\widetilde{H}_{3i-2,3i-2}^{-1} + \widetilde{H}_{3i-1,3i-1}^{-1} + \widetilde{H}_{3i,3i}^{-1} \right] \qquad (47)$$

Additionally, the cross correlation between different residues can be calculated via:

$$\langle \Delta R_i \cdot \Delta R_j \rangle = \langle \Delta X_i \cdot \Delta X_j \rangle + \langle \Delta Y_i \cdot \Delta Y_j \rangle + \langle \Delta Z_i \cdot \Delta Z_j \rangle$$

$$= \frac{k_B T}{\gamma} \left[\widetilde{H}_{3i-2,3j-2}^{-1} + \widetilde{H}_{3i-1,3j-1}^{-1} + \widetilde{H}_{3i,3j}^{-1} \right] \qquad (48)$$

Anisotropic network models were generated for each protein with a cutoff of 15Å for alpha carbon interactions calculating 20 normal modes per protein structure. Eigendecomposition was computed to determine the mean square fluctuations of alpha carbons, elastic potential energy and B-factors using the DSSP module in MATLAB as previously published. [211] Fluctuation data was used to generate machine learning features while fluctuations, elastic potential energy, and B-factors were all factored into molecular modeling and simulation analysis. All model visualization, movie trajectories, and image rendering were completed in Jmol. [212] All programing methodology is further detailed in **Appendix B**.

3.5 Structural analysis using molecular modeling and simulation

All protein and ligand structural data was obtained from the PDB. [204] Receptors lacking solved structure data were omitted from all datasets and analyses. All structural analysis was completed in the UCSF Chimera 1.11.2 program. [213] Atomic contacts and hydrogen bonding prediction were completed post-structure steepest descent minimization and solvation in a TIP4P-Ew model. Solvation models are crucial to calculation accuracy as solvation energies and intermolecular interactions between solvents and solvated molecules will affect all other interaction calculations. [82]The TIP4P-Ew model is a re-parameterization of the TIP4P model using Ewald techniques. The TIP4P water solvation model is a rigid planar four-site interaction potential for water consisting of a Lennard-Jones site for the oxygen atom and three charge sides with a 'mock' atom maintaining molecule geometry and force distribution. [214] The TIP4P model is summarized in **Table 3**.

r_{OH} (Å)	HOH , deg	σ (Å)	ε/ k (K)	q(O) (e)	q(H) (e)	q(M) (e)	r_{OM}(Å)
0.9572	104.52	3.154	78.0	0	0.52	-2q(H)	0.15

Table 3. TIP4P Model Parameters. The TIP4P model is described above in terms of bond angles and radii, dielectric constant, and charge. TIP4P contains a Lennard-Jones site for the oxygen atom where ε describes the well depth and σ is the value of r at which the Lennard-Jones potential equals zero.

When adjusted for the Ewald sum technique to better simulate electrostatic interactions within a system via the introduction of charge "clouds", the model is adjusted to values as described in **Table 4.** A visualization of the TIP4P-Ew model is given in **Figure 14**.

r_{OH} (Å)	HOH , deg	σ (Å)	ε (kcal.mol^{-1})	q(O) (e)	q(H) (e)	q(M) (e)	r_{OM}(Å)
0.9572	104.52	3.16435	0.162750	0	0.52422	-2q(H)	0.1250

Table 4. TIP4P-Ew Model Parameters. The TIP4P-Ew model is described above in terms of bond angles and radii, dielectric constant, and charge. TIP4P-Ew contains a Lennard-Jones site for the oxygen atom where ε describes the well depth and σ is the value of *r* at which the Lennard-Jones potential equals zero. In contrast to TIP4P, the TIP4P-Ew is reparameterized to include the Ewald sum technique, improving the simulation of long range electrostatic interactions.

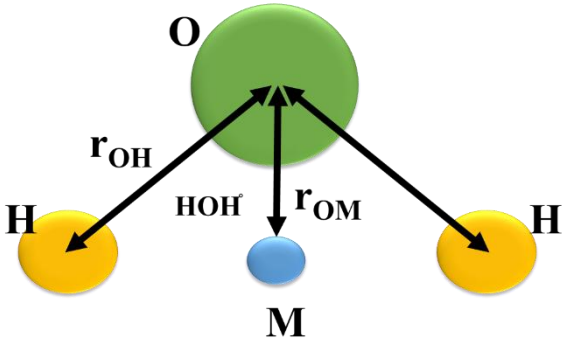


Figure 14. Representation of the molecular model of water used in the TIP4P-EW solvation model. A 'mock atom' is placed in between the hydrogen atoms to maintain model geometry where rOH and rOM are the bond lengths between the oxygen and hydrogen and mock atoms, respectively and HOH* refers to the bond angle of the molecule.

Following structure minimization and solvation, atomic contacts and hydrogen bonds are predicted based on the theoretical calculation of the overlap between two given atoms for all atomic pairs in a given protein receptor where overlap is defined as:

$$overlap_{ij} = r_{VDWi} + r_{VDWj} - d_{ij} - allowance_{ij} \qquad (49)$$

where allowance refers to a correction for potentially hydrogen-bonded pairs in which an allowance > 0 reflects the observation that atoms sharing a hydrogen bond can

have a shorter atomic distance than would be predicted from their van der Waals radii. Allowances are only taken into account for pairs containing a hydrogen donor and an acceptor. Atomic contacts are determined via a cutoff value of -0.4 Å and an allowance value of 0.1 Å. Pairs that are >3 bonds apart are ignored. Using a similar protocol combined with previously published methodology [215], hydrogen bonding pairs are predicted. All models are visualized and images generated using the built-in rendering tools of Chimera 1.11.2. All programing methodology is further detailed in **Appendix B**.

3.6 Logistic regression model parameters

Logistic regression models were built using the scikit-learn class within the linear_model module. (**Appendix B**, [115]) **Table 5** describes the chosen hyperparameter values. Omitted parameters were set to the default values. The SAGA solver is a variant of the Stochastic Average Gradient (SAG) method which optimizes the sum of a finite number of smooth convex functions. SAG converges much faster than other stochastic gradient methods as its iterative cost is independent of the number of function terms in the calculated sum. While it is faster than other solvers, it can be impractical for large *n* due to its memory cost. The SAGA variant differs in that it also supports non-smooth regularization options, making it a more ideal choice for sparse and/or very large datasets. The scikit-learn documentation recommends the SAGA solver as the best choice in most cases despite limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) being the default option. [115]

penalty	l2
class_weight	balanced
solver	saga

Table 5. Logistic Regression Model Hyperparameters. Optimized hyperparameters for all logistic regression models. Omitted hyperparameters were not adjusted from default values.

3.7 Decision tree-based model parameters

Decision-tree based classifiers including random forest, extra trees, and bagging decision tree models were constructed using the scikit-learn ensemble module. (**Appendix B,** [115]) The extra trees classifier compares to a random forest classifier in that they each build multiple decision trees in which nodes are split using random subsets of features. The two algorithms contrast in that extra trees classifiers do not bootstrap observations and nodes are split randomly rather than on best splits. Bagging trees differs from random forest in that all features are taken into account when splitting a node rather than randomly selecting a subset of features. Optimized hyperparameters for each method are summarized in **Table 6**. Omitted parameters were set to default values.

Model	Random Forest	Extra Trees	Bagging Trees
max_features	sqrt	sqrt	1.0
n_estimators	1000	1000	1000
bootstrap	True	True	True

Table 6. Decision Tree-Based Model Hyperparameters. Optimized hyperparameters for all decision tree-based models, including random forest, extra trees, and bagging trees classifiers. Omitted hyperparameters were not adjusted from default values.

3.8 Support vector machine model parameters

Support vector classifiers were built using the scikit-learn ensemble module. (**Appendix B,** [115]) **Table 7** describes the optimized hyperparameter set used. A polynomial kernel was selected over the more common radial basis function (RBF) due to its increased computation speed without sacrificing performance (*data not shown*).

C	0.6
kernel	poly
gamma	auto

Table 7. Support Vector Machine Model Hyperparameters. Optimized hyperparameters for all SVM models. Omitted hyperparameters were not adjusted from default values.

3.9 AdaBoost model parameters

AdaBoost classifiers were built using the scikit-learn ensemble module. (**Appendix B,** [115]) Default parameters were found to result in the optimized model and can be found in the *sklearn* documentation.

3.10 Performance metrics and cross validation methods

All model performance is measured by a combination of precision, recall, confusion matrices, F1-scoring, and ROC AUC scores as described in Chapter One.

Each test set performance is measured by a 10-fold cross-validated F1 score implemented via the scikit-learn metrics module. (**Appendix B,** [115]) In K-Fold Cross Validation, data is randomly shuffled and split into *k* numbers of folds (i.e., groupings of data). One fold is separated into the test dataset, while the other *k*-1 folds are grouped together into the training data. The model is fitted and evaluated via a chosen performance metric score. The process repeats iteratively with each fold as the test data and the final *k* scores are averaged to form a more robust model validation score. F1-scoring was chosen as the performance metric due to the binary nature of the models as well as the consequence of positive class errors. [216]

3.11 Deep neural network model parameters and architecture

Models were replicated using publicly available code from GitHub repositories provided by authors. All models were replicated and tested in either a GPU-enabled Python 3 environment using JupyterLab or within the SageMaker service from Amazon Web Services (AWS).

A deep feedforward neural network was built within the PyTorch framework. [217] Despite experimentation with more complex architectures, it was found that a feedforward architecture was sufficient to achieve high accuracy and ideal over other methods due to its computational speed. The model architecture, summarized in **Figure 15**, is composed of five linear layers with batch normalization following each layer and dropout layers following the first and last hidden layers. Batch normalization was implemented to increase model speed and stabilize each layer's input via re-standardization of the data whereas dropout layers were implemented to prevent overfitting. [139], [169] Placement of batch normalization and dropout was guided by previously published architectures. [21], [133], [169] Model implementation with the Exponential Linear Unit (ELU) activation function was found to result in better performance than other activation functions. ELU activation is more computationally expensive than the popular Rectified Linear Unit (ReLU) activation function, but in addition to avoiding the vanishing gradient problem, it also avoids the dead ReLU

problem. Loss is measured by the binary cross entropy criterion. Performance metrics include cross validated F1-score, precision, recall, and ROC AUC score.

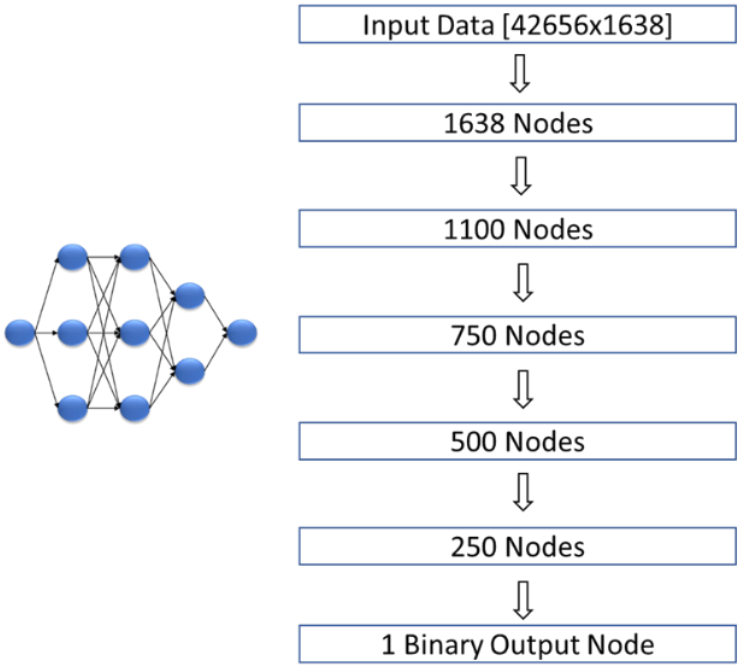


Figure 15. Architecture of feedforward, fully-connected deep neural network used for drug-target interaction prediction of Subgroup A17 receptors. A model with five hidden layers (each followed by batch normalization and a dropout layer following the first and fifth hidden layers (p=0.2 and p=0.7 respectively) was found to result in the best performance.

4. Results and Discussion

4.1 Specific Aim 1. To test the performance of current machine learning algorithms for the classification of protein binding behavior and determine the extent to which these architectures can be applied to the classification of Class A GPCR-drug interactions.

4.1.1 Current state-of-the-art method performance against a DUD-E benchmark dataset

To validate the replicability of the selected prior published protein-ligand interaction neural network classifiers (DeepConv [21], SSNet [22], DeepScreen [20], and DeepAtom [19]), trained models were built from publicly available code provided by the authors in each model's respective framework as written. Each publication uses different performance metrics; however, for consistency in our analysis we chose to measure accuracy via the F1-score metric. While this makes direct comparison to published results less interpretable, generalized conclusions on model accuracy are still possible. Multiple, but not all, of the models selected the DUD-E database as a validation set but all models were tested on a generalized protein-ligand database containing a wide breadth of protein families and ligand classes, therefore the DUD-E database is a suitable test set for all four models. A 10-fold cross validated F1-score (± standard error) for each model is given in **Figure 16**.

DeepConv reports an AUC score of 0.852 [21] against a dataset constructed from a mixture of drug-target interaction databases (e.g., KinaseSAFARI, PubChem, DrugBank) while our model obtained a cross-validated F1-score of 0.764 ± 0.028. (**Figure 16**) It is of note that our DUD-E test set was considerably smaller than the test set from published results (438 interactions vs. >36,000 interactions) which potentially influenced results.

DeepScreen reports an F1-score of 0.87 [20] against a large dataset of experimentally-validated protein-ligand interactions, and our replicated model results were in strong agreement with these published results. The replicated pre-trained model resulted in a cross-validated F1-score of 0.88±0.016 against the DUD-E test set. (**Figure 16**)

DeepAtom reports a Pearson correlation of 0.83 [19] against a dataset of PDBind protein-ligand complexes. When the DUD-E test set was applied to the replicated pre-trained model, an F1-score of 0.805±0.0304 was obtained. (**Figure 16**)

SSNet reports an AUC of 0.93 [22] against a dataset containing protein-ligand interaction data from DUD-E with K_d values greater than 10,000nM or smaller than 100nM. When applied to our broader range of DUD-E test data, the model resulted in a cross-validated F1-score of 0.81±0.075. (**Figure 16**)

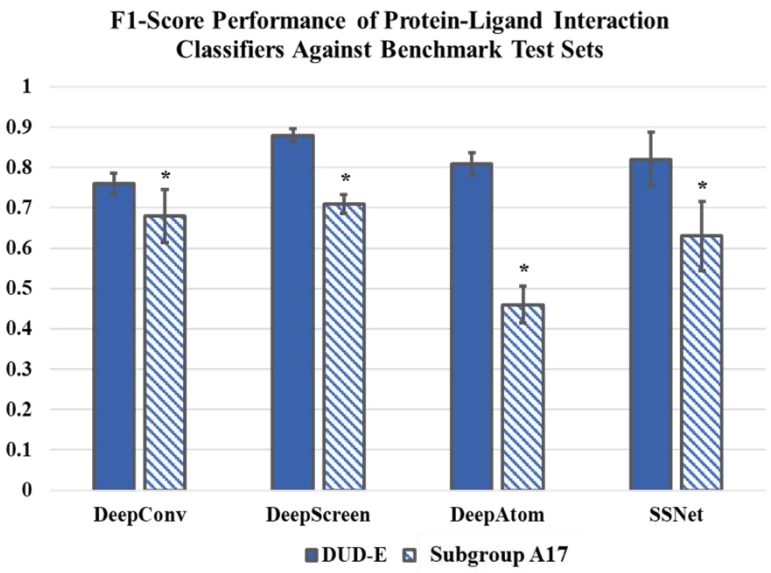


Figure 16. Protein-ligand interaction classifiers perform significantly worse than reported accuracy metrics on a test set containing Subgroup A17 GPCRs. Model performance of four previously published architectures against a benchmark test set adapted from the DUD-E interaction dataset and a test set consisting of drug-target interactions from the Class A Subgroup A17 of proteins including dopamine and serotonin receptors. Accuracy was measured by a 10-fold cross validated F1-score. Each replicated model performed significantly worse against the Subgroup A17 dataset than the DUD-E dataset.

4.1.2 Current state-of-the-art method performance against a GPCR Subgroup A17 test set

Despite the potential concern associated with the DeepConv author's dataset creation methodology, the accuracy of this model was still comparable to published metrics when tested against the experimentally validated DUD-E dataset. However, once applied to a specific protein family test set, model accuracy significantly dropped ($p<0.05$). This additional test set contained experimentally validated data for proteins within the GPCR Subgroup A17 (serotonin and dopamine receptors) and resulted in in an average cross-validated F1-score of 0.685 ± 0.05, both a lower accuracy and higher standard error rate than that of the more general DUD-E dataset. (**Figure 16**)

While accuracy did significantly drop for the Subgroup A17 test as compared to the DUD-E test set ($p<0.05$), the model against the Subgroup A17 data still obtained a cross-validated F1-score of 0.71 ± 0.028. (**Figure 16**)

The DeepAtom framework performed well against the general DUD-E dataset but did not demonstrate any predictive ability against the Subgroup A17 dataset, as evidenced by the cross-validated F1-score of 0.462 ± 0.043. This data suggests that the structural features extracted from the serotonin and dopamine receptor complexes via the DeepAtom algorithm are not useful for interaction prediction. (**Figure 16**)

SSNet performed significantly worse against the Subgroup A17 dataset than the general DUD-E test set ($p<0.05$) with a comparably large standard error (cross-validated F1-score of 0.62 ± 0.088). (**Figure 16**)

4.1.3 Discussion of current published models and potential applicability to Subgroup A17 proteins

While we hypothesized that training on a broader, more general dataset of protein-ligand interactions would result in a higher test accuracy for a specific protein group due to similar overarching physiochemical properties that determine protein binding behavior and a larger volume of available training data, this was not observed for any of the replicated classifiers. It is possible that while the available amount of data for generalized protein-ligand interactions is larger, the conformational dynamics governing ligand binding behavior is too different between groups with drastically different structure to simplify structure-function relationship for ligand binding across multiple protein families. This data suggests that the potential for transfer learning within protein-ligand binding is lower than expected.

A strength of DeepConv, as stated by the authors, is the generable applicability of the model due to the model only requiring sequence data and no structural information. [21] However, this data suggests that despite not using three-dimensional structures, the applicability of this model is limited. It is also of note that while the DeepConv model does not require structure information, the input format requires available ChEMBL data. While this program feature was likely introduced to improve user experience, it does limit the data compatibility range of the model.

The Subgroup A17 dataset contained 52 observations and it is possible that the small dataset size influenced model accuracy. It is a noted limitation of this aim that finding data to satisfy the input requirements of all four models significantly reduced the availability of usable data. While it is possible to modify these models to allow different input structures or re-train the model using our datasets, the amount of work necessary to complete these tasks would be overambitious and beyond the scope of this project.

The DeepScreen authors' data collection strategy for both training and testing data is an additional factor that may address disparities between the published accuracy of DeepScreen and accuracy against other datasets. [20] For the datasets tested in this work, we chose to only include experimentally validated negative interactions. Additionally, test data were randomly generated from the full dataset using the train_test_split function from the scikit-learn model_selection module. [115]

In contrast, the data used to train and test DeepScreen by the authors contained experimentally validated positives and randomly generated 'scrambled' negative data (i.e., positive interaction pairs were randomly shuffled to generate a new set of theoretical interaction pairs and any overlapping samples between the original and shuffled dataset are removed to generate a novel dataset containing new interactions which will be labeled as negatives). These interactions were not compared to published experimental data. As the dataset was pulled from a small subset of the publicly available protein-ligand interaction data, it is very likely that a large number of experimentally-validated positive interactions were not included in the data and therefore it is also possible that some of the scramble-generated 'negative' interactions have been experimentally validated as positive, or not yet tested to confirm either way. Moreover, the data chosen to be

scrambled by the researchers was hand-selected and chosen by protein family rather than a random distribution of all data. As a result, researcher bias may have been introduced into the dataset. It is possible that rather than predicting drug interaction probabilities, the program was identifying real vs. randomized data observations.

Overall, the DeepScreen model outperformed other models against both test sets. However, a significant drawback to this model is the requirement of detailed structural information for each interaction. DeepScreen utilizes a three-dimensional apoprotein structure, a ligand structure, and a protein-ligand complex. While the authors were able to compile a large training dataset by including a broad range of proteins from multiple data sources and multiple protein families, this model has limited applicability as solved protein structures are rare and solved protein-ligand complex structures are even more rare. Additionally, the complexity and high dimensionality of this data results in extremely high model complexity and long training times. As a result, despite the strong evidence that DeepScreen displays for the use of structural information in protein-ligand interaction classification, the field of structure-oriented machine learning research still lacks real-world feasibility in terms of general applicability.

The DeepAtom model demonstrates reasonable performance against a general protein-ligand interaction dataset but does not perform well against a test set of family-specific interactions. [19] The DeepAtom model is arguably one of the most complex predictive models published for protein-ligand interaction prediction currently available. Three-dimensional protein-ligand complex structures are voxelized with channels containing encoded feature information describing atoms within each respective voxel. The voxelized structure undergoes a three-dimensional convolution to predict an

inhibition constant value. Despite the complexity of the model structure, only 24 features are used to describe the entire complex and the results of our analysis suggest that these 24 features are not generally applicable to all protein-ligand interaction datasets.

SSNet resulted in a comparable average F1-score for the DUD-E data set and the Subgroup A17 data set to the other three tested models. [22] The standard error for this model was considerably higher than other models and unusually high for a cross-validated classifier performance, suggesting that a certain amount of randomness is contributing to the model prediction. In addition to the concern provided by this large standard error repeated across multiple experiments (data not shown), the model was only trained to handle extreme values for positive interactions. It is possible that the model does not predict well for positive interactions with Kd values above 100nM as the training data only classified interactions with dissociation constants below this threshold. As an example, the D2 receptor has a reported affinity for dopamine ranging between ~400-700nM. [26] Under the classification threshold criteria for SSNet, dopamine would not be considered a positive ligand for D2. The lack of consistency and range of ligand input of the SSNet predictive ability raises concern about the model's potential for applicability.

Overall, while each tested model demonstrated strengths to guide the development of future models, each also raised concerns about classifier replicability and performance generalizability. Each model performed relatively well compared to other published classification models in the literature against the DUD-E adapted test set, but no models transferred well to the Subgroup A17 dataset. These results not only demonstrate a need for improved models to be developed, they also provide valuable information to guide

future directions. Firstly, the poor performance against Subgroup A17 data suggests that the hypothesis that general relationship patterns in protein-ligand binding can be universally applied to more specific protein groups is false. Although it would be ideal to be able to train on a larger, broader data set to maximize learning potential and intermolecular interactions are generally governed by the same physical and chemical principles, it appears that the biochemical conformational dynamics responsible for ligand binding are too complex and unique within protein classes and subgroups to be applied broadly. To determine feature importances for Subgroup A17, it would be possible to iteratively freeze all model weights with one feature left out to better characterize relationship between each feature and output accuracy.

4.2 Specific Aim 2. To assess the degree with which model performance of a Class A GPCR subgroup classifier can be maintained as we reduce time complexity via structure-based feature selection.

4.2.1 Multiple structural alignment of Class A GPCRs

A dataset of 29 Class A GPCR protein structures from a range of subgroups was collected and aligned using the Needleman-Wunsch algorithm within the Caretta structural analysis program, based on the BLOSUM62 similarity matrix for the twenty standard amino acids. [192], [218] After gaps in the alignment are omitted, 174 residues are aligned. Only fully aligned residues are included in any comparative studies to avoid handling of missing values. A model visualization of the superimposed structures is given in **Figure 17**.

In addition to the multiple-subgroup Class A dataset, two additional alignments were generated for the structural alignment of four serotonin receptors (**Figure 18**) and four dopamine receptors (full visualization not shown; binding site alignment shown in **Figure 19**). The alignment of serotonin receptors (5-HT1A, 5-HT1B, 5-HT2A, 5-HT2C) reflected currently published data demonstrating that residues within the N-terminal loop and ECL3 regions are highly conserved between serotonin receptors. The alignment of dopamine receptors (DRD1, DRD2, DRD3, DRD4) reflected currently published data demonstrating that twelve out of thirteen key ligand-binding residues are highly conserved between dopamine receptors. [5] For additional comparative structural

analyses, all 37 Class A proteins (multiple-subgroup Class A dataset, serotonin receptors, and dopamine receptors) were aligned to determine structurally-unique regions for the serotonin and dopamine subgroup within the context of each other and the overall structural landscape of Class A GPCRs.

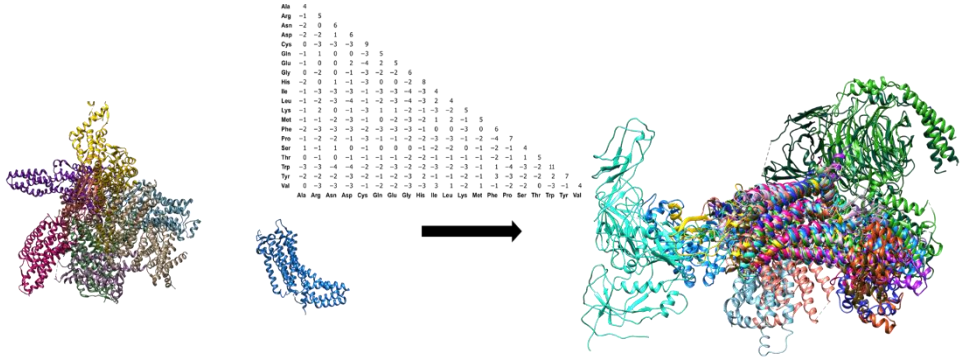


Figure 17. Multiple structural alignment of Class A GPCR proteins. 29 proteins, each in different color, are randomly positioned in three dimensional space (left) and aligned into one overlapping structure (right). Alignment was created using the Needleman-Wunsch algorithm based on the BLOSUM62 similarity matrix. Following alignment and removal of gaps between all proteins, 174 aligned residues remained.

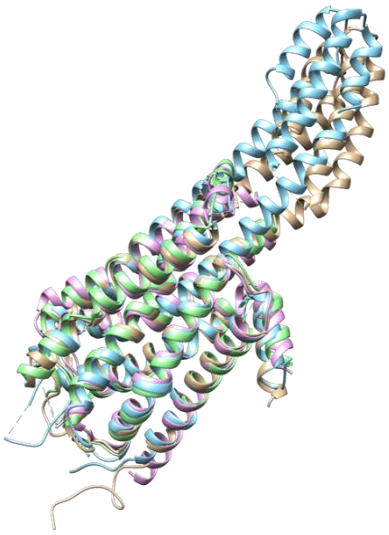


Figure 18. Multiple structural alignment of serotonin receptors 5-HT1A (PDB 7E2X), 5-HT1B, (PDB 5V54) 5-HT2A (PDB 6A94), and 5-HT2C (PDB 6DS0). Alignment was generated using the Needleman-Wunsch algorithm. High structural similarity reflects previously published data demonstrating binding site similarity between Class A GPCRs, reducing effectiveness of feature sets focused on binding site residues.

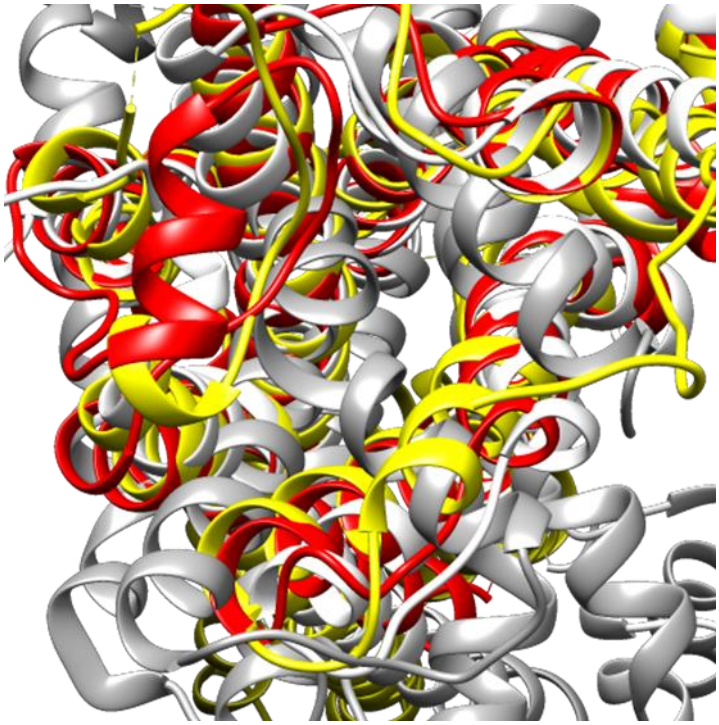


Figure 19. Multiple structural alignment of the D2 (PDB 6VMS)/D3 (PDB 6CMU)/D4 (PDB 5WIV) receptors focused on the binding site. Alignment was generated using the Needleman-Wunsch algorithm. High structural similarity reflects previously published data demonstrating binding site similarity between Class A GPCRs, reducing effectiveness of feature sets focused on binding site residues.

4.2.2 Anisotropic network model analysis of Class A GPCR proteins

For all 29 proteins within the multiple-subgroup Class A dataset, all four serotonin receptor structures, and all four dopamine receptor structures included in our analysis, an anisotropic network model was created. As an example, the visualization of the theoretical first normal mode of 5-HT1A is shown in **Figure 20**.

In addition to atomic fluctuations and elastic potential energies (spring forces), it is also possible to calculate the B-factor (also referred to as the temperature factor) of each residue from the ANM. The B-factor quantitates the uncertainty of each atom under the assumption that uncertainty in atomic position increases with disorder in the protein structure. B-factors are directly correlated with the degree to which the electron density for a given atom is spread out. A high B-factor reflects a low empirical electron density for the atom and vice versa for low B-factors. While B-factors can also be used to describe model error from X-ray crystallography, within thermodynamics theory, the B-factor represents the dynamic mobility of an atom. The B-factor is given by:

$$B_i = 8\pi^2 U_i^2 \qquad (50)$$

where U_i^2 is the mean square fluctuation for a given atom *i* and as *U* increases, the B-factor increases proportionally. B-factor can be useful when analyzing the contribution of a given residue to the overall conformational dynamics of a protein and conformational dynamics have been demonstrated to play an important role in ligand association and dissociation. [219], [220]

Within the ANM analysis for included Class A GPCR proteins, solvent-accessible regions demonstrated higher B-factors than membrane-embedded regions.

Using structural data from previously calculated multiple structural alignments in combination with ANM fluctuations for the alpha carbon of each residue for each protein, it is possible to compare atomic fluctuation data at each aligned residue position between all Class A GPCR proteins included in this study. The variance at each aligned position was calculated and residue positions were enumerated by magnitude of variance.

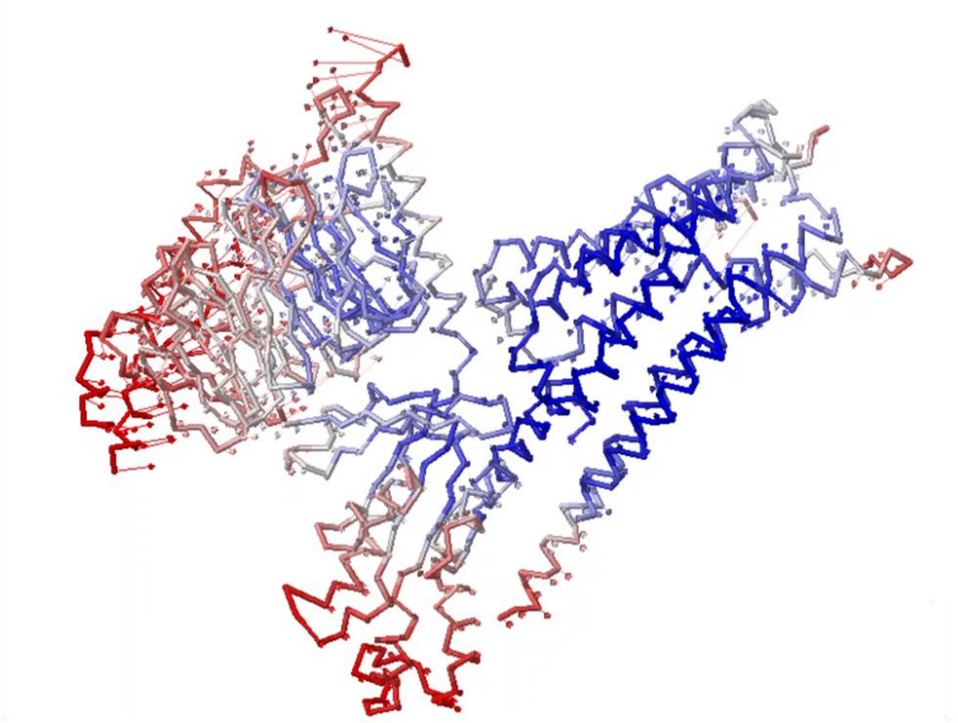


Figure 20. Visualization of 5-HT1A Anisotropic Network Model, normal mode 1 out of 20 normal modes generated. Arrow vectors indicate theoretical direction and magnitude of atomic fluctuations for the specific represented mode. Model color gradient represents B-factor where red corresponds to higher values and blue corresponds to lower values.

4.2.3 Comparative structural analyses of Class A GPCR binding behavior and discussion of potential evidence provided by ANM data

Following minimization and solvation of apoprotein receptors, specific residues identified by previous published structural analysis or our ANM results were examined to determine what extent the ANM provided novel information. For protein-ligand complex structures, predicted intermolecular hydrogen bonding between receptor and ligand were calculated for all potential donor/atom pairs via atomic contact modeling. **Figure 21** shows the structure of rhodopsin with functionally-important residues identified by published studies in blue, ANM-identified regions of high structural deviation (i.e., regions in which theoretical atomic fluctuation patterns differ strongly from other Class A proteins) compared to other Class A GPCRs in red, and any overlapping residues previously identified that were also identified as structurally unique by ANM in green. Outside of a single residue, all residues highlighted by ANM were novel and not identified by our literature analysis of rhodopsin structure and conformational dynamics. ANM-identified residues of high structural deviation had a strong positive correlation with reported B-factors from the original published X-ray crystallography data. [221] This correlation provides additional evidence that the ANM-identified regions are likely to be influential in rhodopsin conformational dynamics.

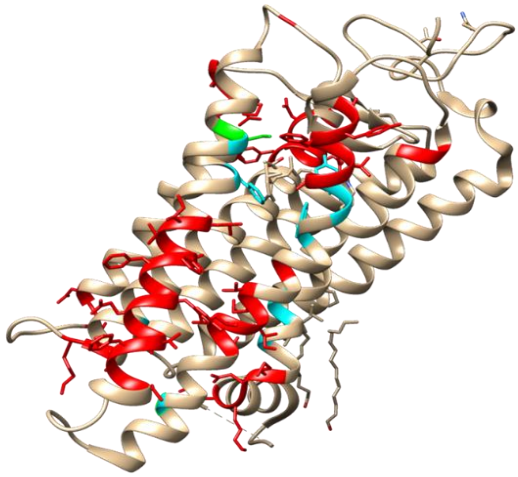


Figure 21. Results from the rhodopsin ANM yield a different residue set than other structural analysis methods. Residues colored in red are regions of high structural deviation identified by ANM analysis. Blue residues are previously identified regions of functional importance. A single residue was identified by ANM analysis that had previously been identified as important by other structural methods. ANM-identified residues highly correlate with X-ray crystallography-obtained residues associated with high B-factor.

Figure 22 represents the binding site of 5-HT1A in complex with LSD, a potent serotonin receptor agonist often recreationally consumed for its hallucinogenic effects. [207] While none of the ANM-identified structurally unique residues of 5-HT1A are predicted to directly interact with LSD, predictive modeling of hydrogen bonding does suggest that two ANM identified residues within close proximity of TM6 form bonds with the toggle switch Trp residue (W6.48) key to the cascade of conformational changes throughout the entire GPCR that result from extracellular ligand binding. [5] Despite the lack of a direct role in ligand binding, this data still suggests that ANM-identified residues of importance are involved in 5-HT1A ligand binding conformational dynamics.

As a final representative case, **Figure 23** displays the binding sites of the D2 receptor (a) and the 5-HT2A receptor (b) in complex with haloperidol, a high affinity dopamine receptor antagonist ($K_i \approx 1.4$nM for D2 receptor-haloperidol interaction [26]). In the bound conformational state, the toggle switch Trp residue is not close enough to the binding site for any hydrogen bonding with the ligand to be predicted. Within the

literature, the behavior of the toggle switch is inconsistent between different receptors within the Class A GPCRs. [222]

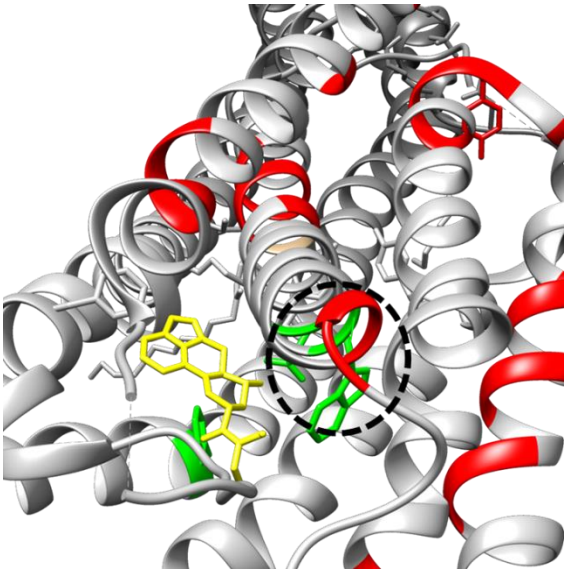


Figure 22. Serotonin receptor atomic fluctuation data identifies important residues outside of binding pocket as evidenced by analysis of the 5-HT1A/LSD complex. Residues identified as possessing high structural deviation via ANM analysis are colored in red. The LSD ligand molecule is represented in yellow and 5-HT1A residues that form bonds with LSD are represented in green. While none of the ANM-identified residues directly interact with LSD, structure analysis within the UCSF Chimera program predicted that hydrogen bonding does occur between two of these residues and a key Trp residue (W6.48; green, circled) for ligand bonding.

While some receptors demonstrate a large conformational change in the W6.48 residue upon ligand binding (formerly believed to be a universal mechanistic change in all Class A GPCR activation), others do not display any significant W6.48 movement. [223] Numerous other toggle switch-based and steric movement-based mechanisms have been identified for various Class A GPCRs. Dopamine receptors have been identified to undergo a W6.48-mediated conformational change upon ligand binding, while serotonin receptors have been demonstrated to undergo a combination of a Tyr toggle switch- and steric hindrance-based mechanism. [222] Moreover, it has been demonstrated that for the 5-HT1B receptor, W6.48 is actually directly involved in ligand binding. [224] While ANM-identified residues within the D2-receptor are not predicted to form any hydrogen bonds with W6.48, two residues within the 5-HT2A receptor are predicted to bond with

W6.48 (as demonstrated for 5-HT1A in **Figure 22**). These hydrogen bonds may potentially influence the conformational dynamics of ligand-mediated Class A GPCR activation via their structural influence on W6.48 toggle location and ligand binding potential.

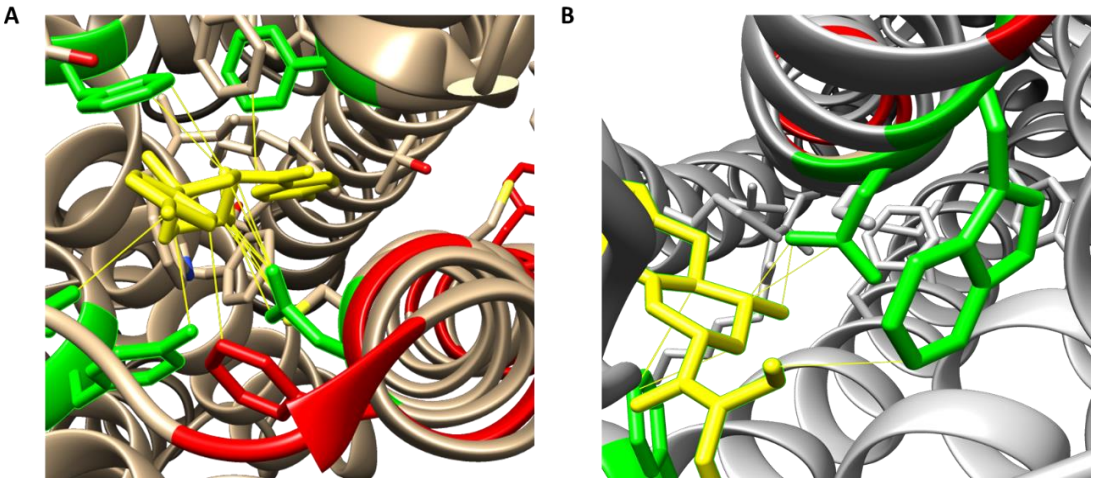


Figure 23. Regions identified as important by ANM analysis highly influence haloperidol binding in psychoactive drug receptors. **A** D2 receptor binding site in complex with haloperidol (yellow). In the bound conformational state, the toggle switch Trp residue (W6.48) is not close enough to the binding site for any hydrogen bonding with the ligand to be predicted. The D2 receptor/haloperidol complex (Ki ≈ 1.4 nM) is predicted to form ten intermolecular hydrogen bonds. Green residues have previously been identified to participate in ligand binding. Red residues have been identified as structurally unique as compared to other Class A GPCRs by ANM and are not predicted to form hydrogen bonds with the toggle switch. **B** 5-HT2A receptor binding site in complex with haloperidol (yellow). In the bound conformational state, the toggle switch Trp is within the binding site and predicted to be actively involved in ligand binding as previously demonstrated with other ligands. The 5-HT2A receptor/haloperidol complex (Ki ≈ 47 nM) is predicted to form four intermolecular hydrogen bonds. Green residues have previously been identified to participate in ligand bonding. Red residues have been identified as structurally unique as compared to other Class A GPCRs by ANM, including two residues predicted to form hydrogen bonds with the toggle switch (Figure 19), pulling it into the binding site.

4.2.4 Influence of structure-based feature selection as determined by ANM on the accuracy and associated time-cost of Class A subgroup classifiers

In addition to the utility of augmented structural modeling analysis with ANM, the literature has also previously demonstrated potential for improved machine learning feature engineering based on normal mode analysis methods including ANM. [192] To test this hypothesis, data generated from multiple structural alignment and ANM of Class A GPCRs was used to build a range of machine learning multiclassification models for subgroup prediction. Due to a strong class imbalance in our 29 protein dataset (majority of proteins belonging to Subgroup 18), SMOTE was applied to the dataset to oversample minority classes. (**Figure 24**)

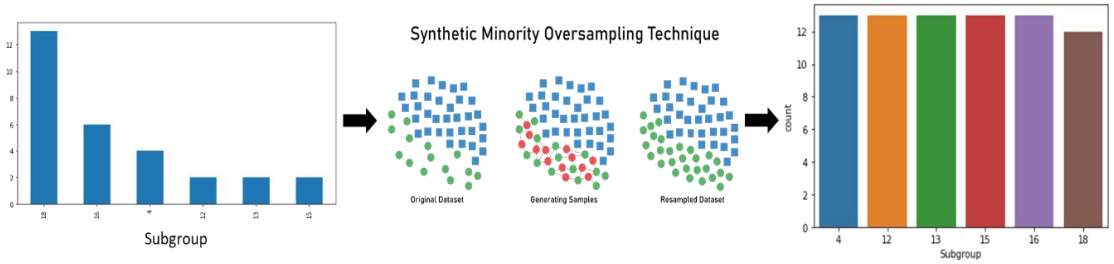
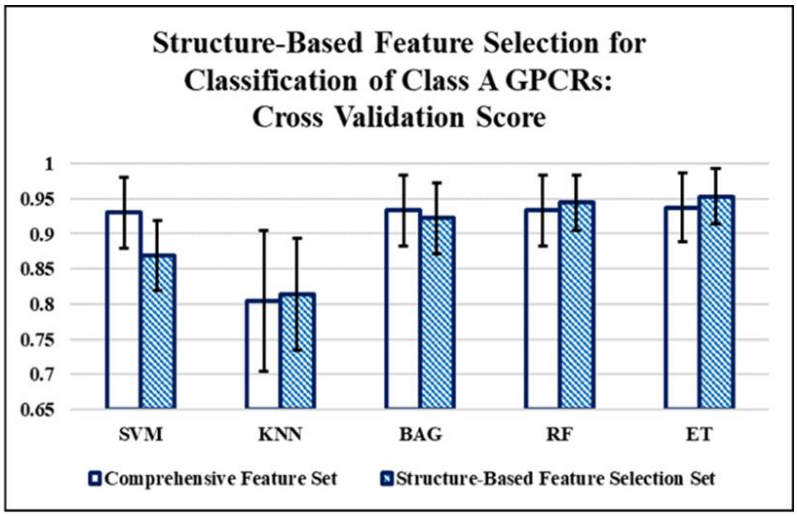


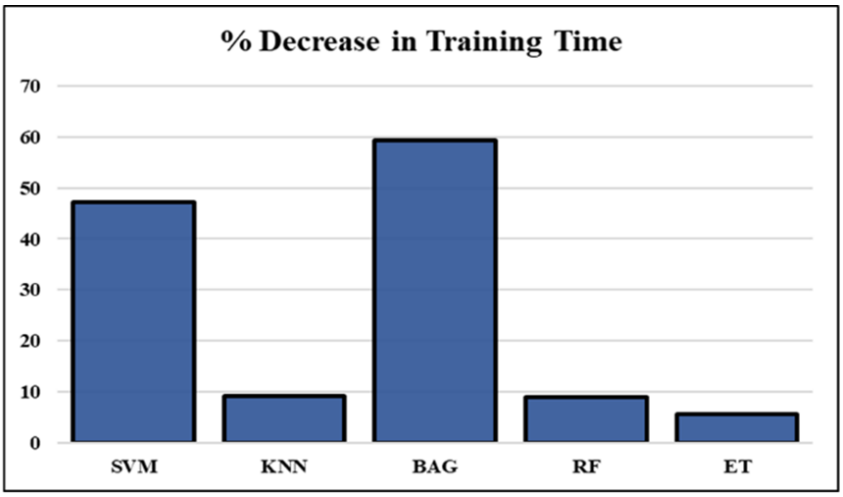
Figure 24. Synthetic data generation to correct class imbalance within the Class A GPCR dataset via SMOTE. Minority classes were oversampled to reach an equivalent *n* to the Subgroup A18 class.

The resulting dataset was randomly split 70% into a training set and 30% into a test set for the building and testing of a range of machine learning models to predict an output class reflective of Class A Subgroup. While multiple models were evaluated, only the results for the top five highest performing models are shown: support vector machine, K-nearest neighbors, bagged trees, random forest, and extra trees. Each model was trained and tested on two separate feature sets—one containing the mean square fluctuation data for each aligned residue position in the dataset, and a reduced feature set only containing mean square fluctuations for the top 15% highest-variation residue positions. These feature sets were contrasted for 10-fold cross validated F1-score performance (**Figure 25)** and training time (**Figure 26**).



	SVM	KNN	BAG	RF	EF
Comprehensive Feature Set	0.93±0.05	0.804±0.1	0.933±0.051	0.933±0.051	0.937±0.049
Structure-Based Feature Selection Set	0.869±0.05	0.814±0.079	0.922±0.05	0.944±0.04	0.953±0.039
Percent Change	-6.5%	+1.2%	-1.1%	+1.2%	+1.7%

Figure 25. Reducing feature set by 85% via structural-based variance threshold does not significantly reduce cross-validated F1-score based-accuracy of classifiers including support vector machine, K-nearest neighbors, bagging trees, random forest and extra trees. (k=10)



	SVM	KNN	BAG	RF	EF
Comprehensive Feature Set (ms)	1.19±0.101	0.34±0.133	7310±36.3	2660±20.7	2290±65.8
Structure-Based Feature Selection Set (ms)	0.628±0.04	0.309±0.065	2970±3.32	2420±10.52	2160±3.08
Percent Change	-47.2%	-9.1%	-59.4%	-9.02%	-5.67%

Figure 26. Reducing feature set by 85% via structural-based variance threshold significantly reduces training time of classifiers including support vector machine, K-nearest neighbors, bagging trees, random forest and extra trees. (k=10)

While we hypothesized that structure-based feature selection methods may potentially increase model classification accuracy from an overall reduction in noise and irrelevant data features introducing bias to the model, we did not see a significant improvement in model performance following variance threshold feature reduction. Model performance was not significantly affected by variance threshold structure-based feature selection in a positive or negative direction for any tested models. We also hypothesized that reducing the feature space via structure-based feature selection would reduce training time for classification models and our results did support this hypothesis. Average percent decrease in training time was significant for each model. Particularly for higher complexity models, egregiously long training times are a major limiting factor for

the real world application and feasibility of machine learning prediction. Although this strategy did not improve model classification accuracy, a decrease in time-associated computational cost without sacrificing accuracy is in itself an improvement in model performance.

It is of interesting note that the random forest model on average had a relatively modest reduction in training time but a slight *increase in* F1-score. While the increase in accuracy was not significant for this experiment, it is potentially possible that a more in-depth optimization of the random forest classifier may further increase this boost in performance and exploratory analysis of the factors underlying this improvement other than decreased training time may potentially result in useful guidance for future work. Moreover, further analysis of the relationship between predictive model architecture and efficacy of this feature selection strategy is warranted from a computer science perspective, but beyond the scope of the biomedical objectives of this research.

4.3 Specific Aim 3. To implement a novel deep neural network classifier for the binary prediction of drug-binding activity for a dataset of (a) serotonin and (b) dopamine Class A GPCRs.

4.3.1 Implementation of structure-based feature selection methods for deep neural network classifiers

While data collected in Specific Aim 2 demonstrates the potential for structure-based feature selection methods, the overall typical training time for the tested models is relatively reasonable for most application scenarios as compared to the training time cost of a neural network, which can span from hours to days. Additionally, the classification task was comparatively simple to an interaction prediction between a drug-target interaction pair—particular when dealing with numerous ligands per the same target receptor, and vice versa, resulting in redundant data features between observations.

We hypothesized that the application of structure-based feature selection methods to deep neural networks trained for the classification of Subgroup A17 drug-target interactions will reduce the number of unnecessary features and as a result, reduce training time as previously evidenced in simpler architectures. Following the construction of a deep neural network classifier as described in **3.13**, models were trained for 500 epochs on a dataset of 5-HT drug-target interactions containing data features describing both the receptor and ligand molecular structure. In addition to the fully comprehensive feature set containing feature information for all receptor residues, a receptor feature set

reduced by 85% via variance threshold selection (as previously described in section **4.2**) was also used to train the classifier to compare the extent to which we can reduce the computational cost of the model via structure-based feature selection without sacrificing classification accuracy. For each training set, only the receptor features were reduced and the same ligand feature set was used. Reduction of features by 85% (features describing 42 residues as compared to 282 residues) did not result in a significant reduction in classification accuracy as measured by 10-fold cross validated F1-score or ROC AUC score. (**Figure 27**) However, this feature reduction method did result in an 85% reduction in training time for the classifier. (**Figure 28**)

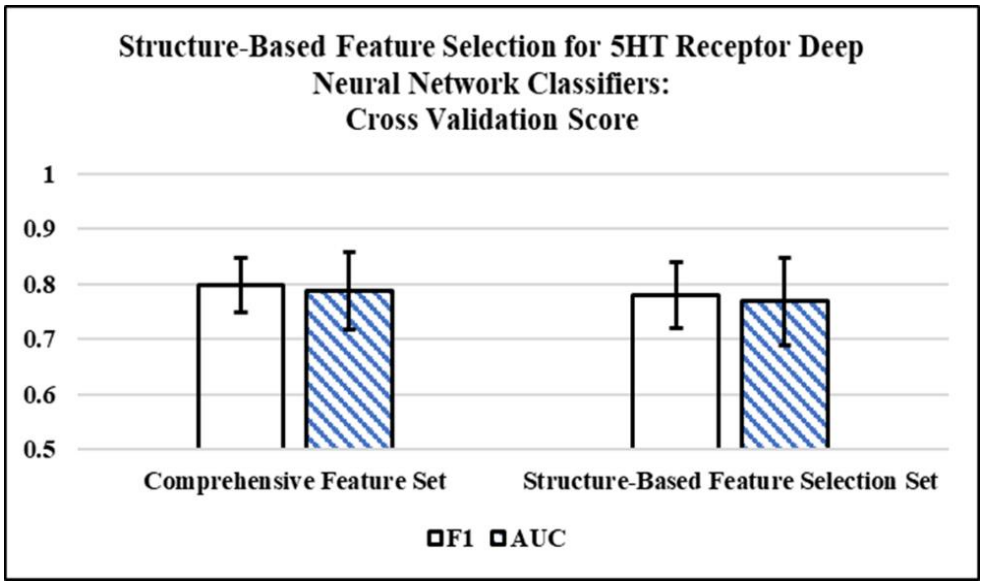


Figure 27. Deep neural network classifier prediction of serotonin receptor activity accuracy was not significantly reduced when the receptor feature set is reduced by 85% as determined by a 10-fold cross validated F1-score or ROC AUC score.

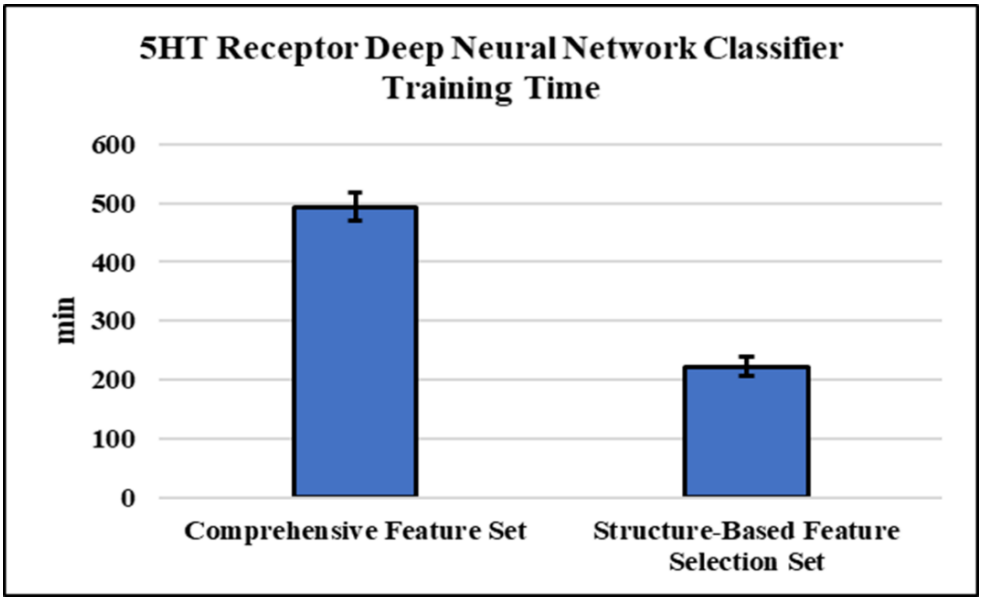


Figure 28. Training time for a deep neural network classifier of serotonin receptor activity was significantly reduced by 45% when the receptor feature set is reduced by 85%.

4.3.2 Deep learning classification of 5-HT receptor drug-target activities

Figure 29 shows the accuracy of a deep fully-connected feedforward neural network against a large dataset of serotonin receptor drug-target interactions following structure-based feature selection methods to predict a binary output of active/inactive per interaction pair. Deep learning accuracy was contrasted to other commonly applied machine learning algorithms on the same dataset. Classification accuracy measured via 10-fold cross validated F1-score and ROC AUC score was significantly improved ($p < 0.01$) in the deep neural network (F1-score = 0.78, AUC = 0.77) as compared to other models. Random forest and support vector machine classifiers also performed well (approximately 10% lower accuracy than the deep learning model) while logistic regression and AdaBoost algorithms had little to no predictive ability for 5-HT receptor interaction behavior. (**Figure 29**)

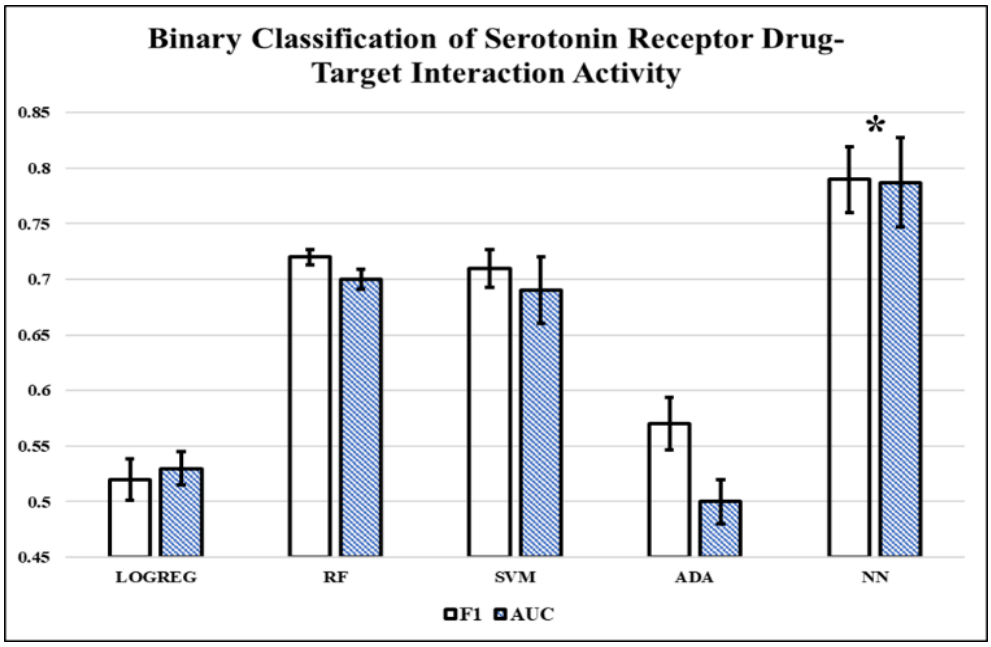


Figure 29. Deep neural network classifier substantially outperformed other classifiers (F1 = 0.78, AUC = 0.768) for prediction of serotonin receptor binding behavior, including logistic regression, random forest, support vector machine, and Adaboost as measured by cross-validated F1-score and cross-validated ROC AUC. (k = 10)

4.3.3 Deep learning classification of DRD receptor drug-target activities

The same network architecture was trained on dopamine receptor drug-target interactions following structure-based feature selection and demonstrated significant performance superiority ($p < 0.01$; F1-score = 0.93, AUC = 0.941) compared to other models, as well as outperforming model accuracy for serotonin receptor drug-target interaction prediction.(**Figure 30**) An unexpected result of this experiment is that each model, with the exception of the support vector machine classifier, performed better on the dopamine receptor dataset than the serotonin receptor dataset.

Although the dopamine receptor dataset is comprised of the same ANM-generated receptor features and RDKit-generated ligand features and used to build a classifier with the same architecture as the 5-HT dataset and model, average cross-validated F1-scoring was approximately fourteen percent higher for the DRD dataset even in spite of a significantly smaller sample size. In future studies, it is worth investigating the reasoning behind this discrepancy. We suggest that a possible explanation for these results lies in the structural similarity of dopamine receptors as compared to serotonin receptors, which are known to have more structural variation between isoforms. [225] Previous work from other groups and our findings suggest that training a classifier on a group of proteins too

dissimilar from one another increases the difficulty of the learning problem despite allowing for a larger amount of training data. [226]

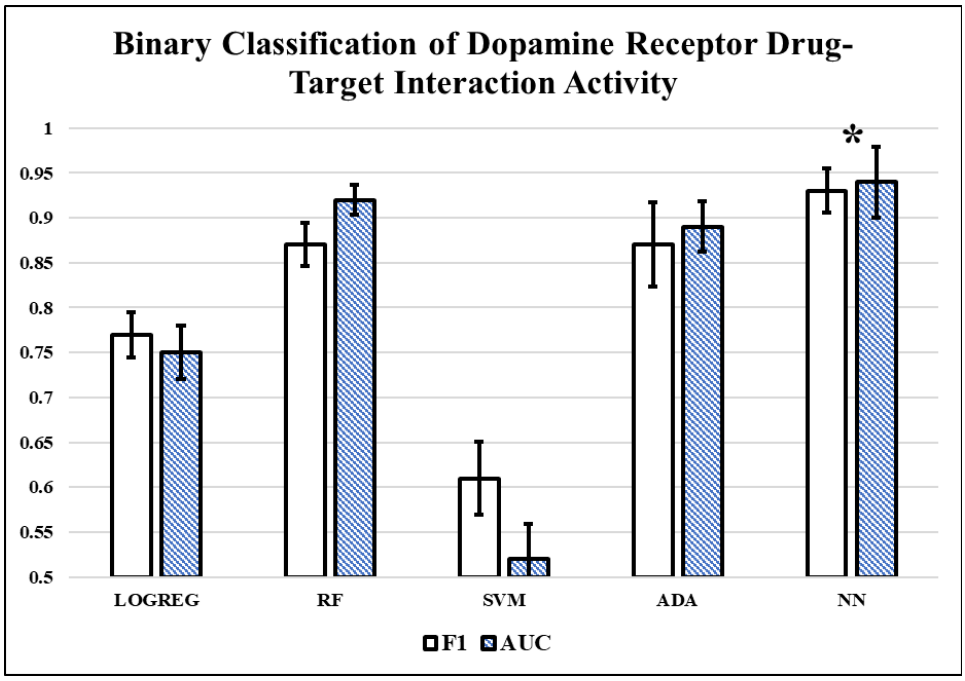


Figure 30. Deep neural network classifier substantially outperformed other classifiers (F1 = 0.93, AUC = 0.941) for prediction of dopamine receptor binding behavior, including logistic regression, random forest, support vector machine, and Adaboost as measured by cross-validated F1-score and cross-validated ROC AUC. (k = 10)

4.3.4 Discussion of machine learning classifier performance for drug-target interaction activities of 5-HT and DRD receptors

Deep neural network performance supported the hypotheses that deep learning strategies will outperform lower-complexity machine learning models due to the high dimensionality of the data and complexity of the research problem and that structure-based feature selection methods will effectively improve the model performance of drug-target interaction classifiers via significant reduction of model training time. While feature selection is a less necessary step prior to model training with neural networks as compared to other machine learning models, our data reflects what has previously been demonstrated the literature for the utility of feature selection in improving computational time and cost for neural network classifiers. [227]

One of the frequently stated benefits of machine learning within computational biochemistry studies is the improved efficiency and handling of big data as compared to molecular dynamics and quantum mechanics trajectories, which can take days to run simulations of a timescale ranging from fs to μs even using parallel computing. [83], [97], [138] However, if the required architecture is highly complex and requires a comparable level of available structural data, the resulting difference in computational cost between simulation trajectories and machine learning models may not be a substantial enough improvement in practice to argue that a meaningful technical innovation has occurred. To compete with state-of-the-art molecular dynamics and quantum mechanics simulation, we

present the argument that a machine learning algorithm for structural biochemistry applications must provide two benefits: a more efficient computational framework and a relatively comparable degree of application interpretability. While molecular dynamics and quantum mechanics methods are computationally expensive, the amount of detailed, interaction-specific structural insight provided is substantially more rich than what is currently available from machine learning strategies.

As an example, consider recent published molecular dynamics studies of fentanyl binding to the μ-opioid GPCR, demonstrating theoretical binding association and dissociation of fentanyl to the μ-opioid receptor binding site with detailed conformational mechanism predictions and novel structural insight. [152] To run these simulations, parallel computing with multiple GPU nodes were necessary to run nanosecond-scale trajectories. Despite the intensive computational power required, molecular dynamics simulation can provide valuable insight into the structure-function relationship of a drug-target interaction.

Deep learning classifiers such as DeepAtom have similar time- and computational-associated costs and require identical amounts of experimentally obtained structure data describing the drug-target complex and each binding partner independently, which can be scarce for the overwhelming majority of protein families. [19] Despite possessing the most complex pipeline of all four architectures included in Specific Aim One's comparative experiment, their reported accuracy is not any higher than the other classifiers and it performed significantly worse against the GPCR Subgroup A17 dataset ($p < 0.01$) than the other classifiers. In reference to the applicability of the knowledge learned by the classifier beyond inhibition constant prediction, the authors provide no

examples and suggest future studies augmenting molecular dynamics simulation with their model. In contrast, DeepConv offers examples of model explainability via demonstrating binding site and class prediction via convolution results per residue. However, as demonstrated by our work presented in Specific Aim Two and published results from other groups, both the binding site recognition problem and the protein classification problem are easily solved by simpler architectures. [30], [199], [228]

While the body of literature has clearly demonstrated the ability of deep learning classifiers and other machine learning models for studying drug-target interactions [19]–[22], [84], it appears that rather than demonstrating performance, justifying utility of these models is the true gap in the literature. Whereas current available models seem to focus on applying the recent state-of-the-art advances in deep learning from a computer science perspective, we observe a relatively comparable performance and cost/benefit trade off of classifiers in a wide range of neural network architectures and both sequence *and* structure-based inputs. While it is of value to take inventory of which computational methods can or cannot be applied to biomedical applications, it is ideal to develop predictive frameworks with the most efficient and accessible architectures where possible.

Perhaps a better objective is to prioritize improving general model applicability via improved feature engineering techniques and improved runtime for computational models. One of the biggest strengths of our proposed deep neural network model is the relative ease of implementation and flexibility of input data formatting requirements. Although numerous architecture types were tested in preliminary experiments (e.g., convolution on ANM data, recurrent backpropagation through time on sequence data;

data not shown), increasing model complexity only marginally improved accuracy while substantially increasing computational cost. As a result, we chose to proceed with a fully connected feedforward network.

Additionally, the use of ANM for feature engineering in our pipeline finds balance between sequence- and structure- based methodologies. While sequence based methods limit performance and explainability, many structure based methods require high computational cost or a volume of available data that is often unrealistic in real-world scenarios (i.e., inactive apostructure and active bound structure for each individual drug-target interaction). Normal mode analysis strategies such as ANM provides a compromise – extracting meaningful, three-dimensional characteristics of a receptor's structure compressed into a one dimensional fingerprint and dynamic behavior without the necessity of complex simulation trajectory. Solved structures for the drug-target complex are not necessary and the receptor features can be generated separately from the ligand features, therefore only the receptor structure is necessary. Because the model allows for fluid range through multiple conformations, the current state or conformation of the experimental structure is not a key factor. While the ANM calculations add model complexity, it is comparatively simpler than many of the discussed alternatives and the predictive classifier is quick to train.

In addition to model efficiency and broad application range due to easily-attainable input data requirements, a key factor in the general applicability of this model is the ease of implementation for a broad range of users. The model architecture is relatively easy to set up with minimal machine learning experience, allowing biochemistry and pharmacology researchers without extensive bioinformatics experience to utilize the tool.

4.3.5 DRD drug-target interaction selectivity prediction for specific clinical applications

Dopamine receptor selectivity majorly influences physiological effects and adverse reactions of dopamine signaling modulating drugs. As an example, behavioral effects and overdose risk from recreational stimulant compounds with differing DRD binding activity display different risk profiles. [229] Benzomethyl ecgonine, or cocaine, is a naturally occurring anesthetic often taken recreationally for its off-label physiological effects including euphoria, increased energy, and lowered inhibition. [230] At higher doses or chronic exposure, adverse effects include restlessness, aggression, hallucination, decreased cardiovascular performance, weight loss, and respiratory damage when administered nasally or inhaled. [39], [231] Unlike other stimulants such as amphetamines, the molecular structure of cocaine does not resemble dopamine and is not a known interacting partner for any GPCR including the dopamine receptors. [26], [206]

A tool that capable of predicting specific activity at the dopamine receptor for dopamine reuptake inhibitor drugs is impactful to the clinical community. Although cocaine does not increase dopamine production, release, or serve as a dopamine receptor agonist, administration of cocaine still results in enhanced dopamine signaling activity via inhibiting the reuptake of dopamine into the nerve terminal via interaction with dopamine transporters, effectively increasing the concentration of cocaine in the synapse available to interact with dopamine receptors. [230]

In contrast, certain ‘designer drug’ stimulants of the cathinone class such as methylenedioxypyrovalerone (MDPV) (also referred to as ‘bath salts’) act simultaneously as a dopamine reuptake inhibitor and a dopamine receptor agonist at the D3 receptor.

[229] As a result, dopamine signaling is upregulated through both increased dopamine concentrations in the synapse and agonism at the receptor, intensifying physiological effects and toxicity risk. MDPV is associated with a higher level of adverse effects than other stimulants including an increased anxiogenic effect, increased aggression, intense, prolonged panic attacks, chronic psychosis symptoms, and higher risk of overdose. [229], [231]

Other designer drugs (i.e., a term typically referring to novel alterations of well known psychoactive compounds with enhanced pharmacological effects developed outside of the realm of medical and legal jurisdiction) in the stimulant category display a similar behavioral pattern of acting simultaneously as a dopamine reuptake inhibitor and a dopamine receptor agonist. [26], [229] Collecting experimental evidence for the relationship between the pharmacological activity profile of designer drugs and their toxicological effects is beyond the scope of this work; however, we hypothesize that one can reasonably infer that dopamine agonist activity in addition to dopamine reuptake inhibitor activity changes the adverse effect profile of a drug.

Within the dataset of psychoactive compounds obtained from the PDSP database, we identified dopamine receptor activity at the D3 receptor, but not the structurally similar D2 or D4 receptors, for designer drugs including MDPV, PPP, methedrone, and PVT. [26] The D2, D3, and D4 isoforms are sometimes grouped separately as their own subtype due to their similar structural and functional characteristics (“Type 2 DRD receptors”), while the D1 and D5 isoforms are considered a separate subtype for their respective similarities (“Type 1 DRD receptors”). [206], [232] One could hypothesize that our deep learning classifier is simply capable of identifying generalized relationships

between the drug structure and its activity against type 1 or type 2 DRD receptors, but due to their structural similarities, cannot differentiate between receptors within the same subtype. To test this hypothesis, we collected a subset of our original DRD receptor dataset for drugs that are selective to either the D2, D3, or D4 receptor (i.e., active at one of the three Type 2 isoforms and inactive at the other two) for a total of 39 observations. Interactions involving the Type 1 DRD receptors or nonselective drugs were omitted from the dataset. Our Type 2 DRD receptor dataset was used as additional test set for the pre-trained model on DRD receptor drug-target interactions. As done previously, predicted output of the model is a binary value corresponding to the activity of the given drug-target interaction pair. The performance metrics for this experiment are given in **Figure 31**.

Figure 31. Accuracy of the deep neural network classifier on a test set of selective DRD drug-target interactions. Precision = 1.0, recall = 0.7857, false positive rate = 0.0, false negative rate = 0.2143, classification accuracy = 0.9231.

Model accuracy is comparable to that of the previous DRD receptor test set, disproving the hypothesis and providing evidence that the model is able to learn isoform-specific structure-function relationships for ligand binding rather than generalizing patterns between isoforms with similar structure.

To further test this hypothesis, we also chose to repeat this experiment with a dataset containing only D2-selective or D4-selective compounds but further reducing the dataset to remove D3-selective compounds resulting in a subset containing 20 observations. This experiment was motivated by the clinical significance of D2 versus D4 receptor selectivity in psychoactive drug behavior. Between 30-60% of patients taking D2 agonists such as antipsychotics report sexual adverse effects including erectile and ejaculatory dysfunction, low libido, and priapism in men and low libido, orgasmic dysfunction, and disturbances to fertility in women. [42], [55], [233]–[235]

In contrast, D4 agonists such as anti-parkinsonism and non-stimulant ADHD therapeutics have a documented potential to increase libido, induce erections in men, and improve vaginal lubrication and orgasm frequency in women. [236]–[238] While currently available sexual function drugs such as Viagra and Cialis are within the PDE inhibitor class, up to 40% of men with erectile dysfunction are non-responsive to PDE inhibitor treatment. [236] Potential sexual dysfunction therapeutics including D4 agonists have been documented in the literature as novel alternatives to PDE inhibition. [235], [236]

Figure 32 summarizes the performance metrics of the deep learning classifier when tested on ability to differentiate between D2-selective and D4-selective compounds. Again, we see a high classification accuracy with this test set. By testing our model on various combinations of receptors, we have demonstrated the model's overall capacity for predicting dopamine receptor drug-target activity as well as dopamine receptor-targeting drug selectivity for highly clinically relevant applications.

Figure 32. Accuracy of the deep neural network classifier on a test set of drug-target interactions selective for the D2 or D4 receptor. Precision = 1.0, recall = 0.8, false positive rate = 0.0, false negative rate = 0.2, classification accuracy = 0.95.

5. Conclusion and Future Work

In summary, we have demonstrated the utility of bioinformatic tools to analyze the relationship between structure and function for protein receptors, providing valuable insight into binding selectivity patterns. Specifically, we have demonstrated the efficacy of these strategies for the Class A GPCRs, and in further detail for the Subgroup A17 GPCRs and the clinical impact of predictive tools for these drug-target interactions. Within the last ten to fifteen years, machine learning models have grown in popularity and promise for the classification of protein-ligand interactions, particularly within the specific investigation of drug-target interaction activity. Recent work has yielded numerous high-performing predictive models for drug-target interactions with some degree of transferrable knowledge to psychoactive GPCR interactions, but improvement is still necessary to improve the general applicability and feasibility of these models. Specific Aim One demonstrates the weaknesses of the current top performing models for protein-ligand interaction as well as raises a question about the value of input feature quality versus model complexity when modeling protein-ligand binding activity. We have presented a novel predictive pipeline combining computational tools from established biochemical modeling techniques, chemical physics simulation, and deep learning for an interdisciplinary framework focused on optimizing not only classification accuracy for predicting drug-target interaction activities, but also model efficiency and practicality.

We have added further evidence to the growing body of literature demonstrating the utility of normal mode analysis methods such as the anisotropic network model in structure-based drug design methodologies. By combining structural analysis with machine learning, we provide structure-function relationship insight that is both accurate and interpretable for GPCR drug-target interactions. While multiple published models have reported high accuracy scoring for protein-ligand interactions, it remains a challenge to extract meaningful information about the conformational dynamics underlying binding activity from machine learning models. While machine learning can provide fast and relatively accurate predictions on the likelihood of a binding interaction to occur, overall, molecular dynamics still remains the superior choice for model explainability. By integrating ANM into our machine learning pipeline as a feature engineering strategy, we sought to strike a balance between the computational efficiency of machine learning prediction and the level of structural detail in molecular model simulations. Results from Specific Aims Two and Three exhibit a range of visualizable structural examples for Class A GPCR interactions relevant to the pharmacology of schizophrenia and mood disorder therapeutics, dementia therapeutics, and substance use disorders.

We hypothesized that a deep learning classifier would be necessary to accurately predict the activity of Subgroup A17 GPCR drug-target interactions due to the complex nature and high dimensionality of the data. This hypothesis was supported for serotonin receptor interactions but other machine learning models generally performed quite well for dopamine receptor interactions. Of note, the support vector machine performed substantially worse than any other models tested for dopamine receptor interaction activity classification. It is possible that the receptor structures are too similar to be

separated using a SVM kernel. In the future, further optimization and investigation is necessary to elucidate the patterns in learning skill demonstrated by different models and data sources in this work.

A significant challenge in completing these experiments lies in the substantial lack of publicly available, reliable experimental data for drug-target interactions. While we chose to use oversampling methods to generate synthetic data for non-real interactions, there are predictive modeling methods available to predict three-dimensional structure for proteins based on sequence, homology, or a combination of both. The validity of these theoretical models is still strongly contested, therefore we chose to only include experimentally validated structures in our dataset. However, it may be possible to include these protein structure predictions in datasets to make predictions for receptors that do not have an experimentally obtained structure. In future directions, it would be of great benefit to determine the effect of training a drug-target interaction classifier with these predicted structures. If performance can be maintained, this experiment would simultaneously present evidence for the validity of computationally generated protein structures and widely increase the range of applicability for interaction classifiers.

In conclusion, we hypothesized that Class A GPCR binding activities can be more efficiently predicted via combining feature engineering and selection based on physiochemical models and deep learning models with subgroup-specific training. In Aim One, we were unable to replicate model performance for pre-trained deep learning classifiers from the literature when tested on a dataset of Subgroup A17 receptor activities. This data provides supporting evidence for our hypothesis suggesting that models trained on a broad protein dataset may not generalize as well to specific protein

subgroups as compared to models trained on datasets containing same-subtype proteins as test data, despite a potentially smaller training set size. Aim Two data demonstrates that the use of feature selection guided by structure can significantly reduce the training time of machine learning models for protein receptor classification problems. Moreover, in Aim Three, we show that this feature selection strategy results in agile training of deep learning classifiers for drug-target interaction activity for two datasets of Class A17 GPCR interactions corresponding to serotonin and dopamine receptors, respectively. The deep learning models generated maintained comparable-or-higher classification accuracy to previously published methods (F1-scores and AUC scores ≥ 0.78) with significantly reduced training time (45% reduction as compared to model architecture trained without feature selection). Cumulatively, this evidence supports the original hypothesis that our proposed framework is capable of maintaining accuracy while improving efficiency of deep learning classifiers for prediction of Subgroup A17 GPCR drug-target interaction activities.

This work establishes a meaningful contribution to the field of bioinformatics within structure-based drug design. Our novel feature engineering and selection methodology based on the unique conformational behavior of a given structure allows for in-depth structural input features without leading to training time exceeding what is feasible for real-world application. Our deep learning architecture is capable of providing an improvement to model accuracy while simultaneously scaling down model complexity to increase the range of applicability and decrease learning curve difficulty for new users.

Within the realm of psychoactive research, low selectivity of therapeutic compounds and unacceptably high rates of toxicity and adverse effects have been a high priority area

of pharmacology research following an overwhelmingly large body of literature suggesting an unacceptable level of risk associated with a range of psychiatric medications that has resulted in a sharp decrease in the number of GPCR-targeting psychoactive compounds receiving FDA approval over the last decade. [18], [75], [80], [239]–[241] Despite this project focusing specifically on GPCR interaction activity within the psychiatric lens, the gained perspective and valuable strategies developed here are highly translatable to any application of protein structure/function relationship analysis. This project serves as a strong foundation for future structure-based drug design and structural analyses across a broad range of clinical applications.

References

[1] W. I. Weis and B. K. Kobilka, “The Molecular Basis of G Protein-Coupled Receptor Activation,” *Annual Review of Biochemistry*. 2018, doi: 10.1146/annurev-biochem-060614-033910.

[2] C. S. Odoemelam *et al.*, “G-Protein coupled receptors: Structure and function in drug discovery,” *RSC Advances*. 2020, doi: 10.1039/d0ra08003a.

[3] R. Fredriksson, M. C. Lagerström, L. G. Lundin, and H. B. Schiöth, “The G-protein-coupled receptors in the human genome form five main families. Phylogenetic analysis, paralogon groups, and fingerprints,” *Mol. Pharmacol.*, 2003, doi: 10.1124/mol.63.6.1256.

[4] H. B. Schiöth and R. Fredriksson, “The GRAFS classification system of G-protein coupled receptors in comparative perspective,” 2005, doi: 10.1016/j.ygcen.2004.12.018.

[5] M. Congreve, C. J. Langmead, J. S. Mason, and F. H. Marshall, “Progress in structure based drug design for G protein-coupled receptors,” *Journal of Medicinal Chemistry*. 2011, doi: 10.1021/jm200371q.

[6] W. K. Kroeze, D. J. Sheffler, and B. L. Roth, “G-protein-coupled receptors at a glance,” *J. Cell Sci.*, 2003, doi: 10.1242/jcs.00902.

[7] Y. Miao and J. A. McCammon, “G-protein coupled receptors: Advances in simulation and drug discovery,” *Current Opinion in Structural Biology*. 2016, doi:

10.1016/j.sbi.2016.06.008.

[8] A. S. Hauser *et al.*, “Pharmacogenomics of GPCR Drug Targets,” *Cell*, 2018, doi: 10.1016/j.cell.2017.11.033.

[9] X. Wang, K. Song, L. Li, and L. Chen, “Structure-Based Drug Design Strategies and Challenges,” *Curr. Top. Med. Chem.*, 2018, doi: 10.2174/1568026618666180813152921.

[10] B. S. Glicksberg, L. Li, R. Chen, J. Dudley, and B. Chen, “Leveraging big data to transform drug discovery,” in *Methods in Molecular Biology*, 2019.

[11] M. Kaur, R. K. Rawal, G. Rath, and A. K. Goyal, “Structure Based Drug Design: Clinically Relevant HIV-1 Integrase Inhibitors,” *Curr. Top. Med. Chem.*, 2019, doi: 10.2174/1568026619666190119143239.

[12] N. M. Rochet, L. A. Kottschade, and S. N. Markovic, “Vemurafenib for melanoma metastases to the brain,” *New England Journal of Medicine*. 2011, doi: 10.1056/NEJMc1111672.

[13] P. D. Leeson and J. R. Empfield, *Reducing the Risk of Drug Attrition Associated with Physicochemical Properties*. 2010.

[14] J. R. Empfield and P. D. Leeson, “Lessons learned from candidate drug attrition,” *IDrugs*. 2010.

[15] C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney, “Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings,” *Adv. Drug Deliv. Rev.*, 2001, doi: 10.1016/S0169-409X(00)00129-0.

[16] J. Ananth, S. Parameswaran, and S. Gunatilake, “Side Effects of Atypical

Antipsychotic Drugs," *Curr. Pharm. Des.*, 2005, doi: 10.2174/1381612043384088.

[17] M. Solmi *et al.*, "Safety, tolerability, and risks associated with first-and second-generation antipsychotics: A state-of-the-art clinical review," *Therapeutics and Clinical Risk Management*. 2017, doi: 10.2147/TCRM.S117321.

[18] H. Y. Wang, C. L. C. Huang, I. J. Feng, and H. C. Tsuang, "Second-generation antipsychotic medications and risk of chronic kidney disease in schizophrenia: Population-based nested case-control study," *BMJ Open*, 2018, doi: 10.1136/bmjopen-2017-019868.

[19] Y. Li, M. A. Rezaei, C. Li, and X. Li, "DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction," 2019, doi: 10.1109/BIBM47256.2019.8982964.

[20] A. S. Rifaioglu, V. Atalay, M. J. Martin, R. Cetin-Atalay, and T. Doğan, "DEEPScreen: High performance drug-target interaction prediction with convolutional neural networks using 2-D structural compound representations," *bioRxiv*. 2018, doi: 10.1101/491365.

[21] I. Lee, J. Keum, and H. Nam, "DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences," *PLoS Comput. Biol.*, 2019, doi: 10.1371/journal.pcbi.1007129.

[22] N. Verma *et al.*, "SSnet: A Deep Learning Approach for Protein-Ligand Interaction Prediction," *Int. J. Mol. Sci.*, 2021, doi: 10.3390/ijms22031392.

[23] A. J. Kooistra *et al.*, "GPCRdb in 2021: Integrating sequence, structure, and function," *Nucleic Acids Res.*, 2021, doi: 10.1093/nar/gkaa1080.

[24] R. Wang, X. Fang, Y. Lu, and S. Wang, "The PDBbind database: Collection of

binding affinities for protein-ligand complexes with known three-dimensional structures," *J. Med. Chem.*, 2004, doi: 10.1021/jm030580l.

[25] M. M. Mysinger, M. Carchia, J. J. Irwin, and B. K. Shoichet, "Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking," *J. Med. Chem.*, vol. 55, no. 14, pp. 6582–6594, Jul. 2012, doi: 10.1021/jm300687e.

[26] NIMH, "Psychoactive Drug Screening Program." https://pdsp.unc.edu/pdspweb/.

[27] B. Roth, W. Kroeze, S. Patel, and E. Lopez, "The Multiplicity of Serotonin Receptors: Uselessly diverse molecules or an embarrasment of riches?," *Neurosci.*, vol. 6, pp. 252–2, 2000.

[28] S. J. Raudys and A. K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Trans. Pattern Anal. Mach. Intell.*, 1991, doi: 10.1109/34.75512.

[29] M. Luo, X. S. Wang, B. L. Roth, A. Golbraikh, and A. Tropsha, "Application of quantitative structure-activity relationship models of 5-HT1A receptor binding to virtual screening identifies novel and potent 5-HT1A ligands," *J. Chem. Inf. Model.*, 2014, doi: 10.1021/ci400460q.

[30] J. V. Hobrath and S. Wang, "Computational elucidation of the structural basis of ligand binding to the dopamine 3 receptor through docking and homology modeling," *J. Med. Chem.*, 2006, doi: 10.1021/jm0501634.

[31] A. A. Kaczor, M. Jörg, and B. Capuano, "The dopamine D2 receptor dimer and its interaction with homobivalent antagonists: homology modeling, docking and molecular dynamics," *J. Mol. Model.*, 2016, doi: 10.1007/s00894-016-3065-2.

[32] V. Zachariou, R. S. Duman, and E. J. Nestler, “Chapter 21 – G Proteins,” in *Basic Neurochemistry*, 2012.

[33] T. K. Attwood and J. B. C. Findlay, “Fingerprinting g-protein-coupled receptors,” *Protein Eng. Des. Sel.*, 1994, doi: 10.1093/protein/7.2.195.

[34] A. S. Hauser, M. M. Attwood, M. Rask-Andersen, H. B. Schiöth, and D. E. Gloriam, “Trends in GPCR drug discovery: New agents, targets and indications,” *Nat. Rev. Drug Discov.*, 2017, doi: 10.1038/nrd.2017.178.

[35] S. Seo *et al.*, “Prediction of GPCR-Ligand Binding Using Machine Learning Algorithms,” *Comput. Math. Methods Med.*, 2018, doi: 10.1155/2018/6565241.

[36] A. Mullard and A. Mullard, “N E W S & A N A LY S I S,” *Nat. Publ. Gr.*, 2016, doi: 10.1038/nrd.2016.136.

[37] K. Sriram and P. A. Insel, “G protein-coupled receptors as targets for approved drugs: How many targets and how many drugs?,” 2018, doi: 10.1124/mol.117.111062.

[38] S. Löber *et al.*, “Novel azulene derivatives for the treatment of erectile dysfunction,” *Bioorganic Med. Chem. Lett.*, 2012, doi: 10.1016/j.bmcl.2012.09.064.

[39] M. Pilla *et al.*, “Selective inhibition of cocaine-seeking behaviour by a partial dopamine D3 receptor agonist,” *Nature*, 1999, doi: 10.1038/22560.

[40] B. A. Ellenbroek, “Schizophrenia,” in *The Curated Reference Collection in Neuroscience and Biobehavioral Psychology*, 2016.

[41] M. M. Sidor and G. M. MacQueen, “Antidepressants for the acute treatment of bipolar depression: A systematic review and meta-analysis,” *Journal of Clinical*

Psychiatry. 2011, doi: 10.4088/JCP.09r05385gre.

[42] D. J. David and D. Gourion, “Antidepressant and tolerance: Determinants and management of major side effects,” *Encephale.*, 2016, doi: 10.1016/j.encep.2016.05.006.

[43] A. Cipriani *et al.*, “Comparative efficacy and acceptability of antimanic drugs in acute mania: A multiple-treatments meta-analysis,” *Lancet*, 2011, doi: 10.1016/S0140-6736(11)60873-8.

[44] D. I. Lubman, J. A. King, and D. J. Castle, “Treating comorbid substance use disorders in schizophrenia,” *International Review of Psychiatry*. 2010, doi: 10.3109/09540261003689958.

[45] K. F. Croom, C. M. Perry, and G. L. Plosker, “Mirtazapine: A review of its use in major depression and other psychiatric disorders,” *CNS Drugs*. 2009, doi: 10.2165/00023210-200923050-00006.

[46] Q. N. Vo, P. Mahinthichaichan, J. Shen, and C. R. Ellis, “How μ-opioid receptor recognizes fentanyl,” *Nat. Commun.*, 2021, doi: 10.1038/s41467-021-21262-9.

[47] J. C. Tolentino and S. L. Schmidt, “DSM-5 criteria and depression severity: Implications for clinical practice,” *Front. Psychiatry*, 2018, doi: 10.3389/fpsyt.2018.00450.

[48] World Health Organization, “Depression.,” 2021. .

[49] American Association of Suicidology, “Some facts about suicide and depression.,” 2019. .

[50] E. A. Madrigal De León, C. I. Astudillo-García, and S. Durand-Arias, “Depression, Substance Abuse, and Suicide Risk: A Rising Problem,” *Addict.*

Disord. their Treat., 2019, doi: 10.1097/ADT.0000000000000155.

[51] J. Haase and E. Brown, “Integrating the monoamine, neurotrophin and cytokine hypotheses of depression - A central role for the serotonin transporter?,” *Pharmacology and Therapeutics*. 2015, doi: 10.1016/j.pharmthera.2014.10.002.

[52] E. H. Lee and P. L. Han, “Reciprocal interactions across and within multiple levels of monoamine and cortico-limbic systems in stress-induced depression: A systematic review,” *Neuroscience and Biobehavioral Reviews*. 2019, doi: 10.1016/j.neubiorev.2019.03.014.

[53] J. M. Ferguson, “SSRI antidepressant medications: Adverse effects and tolerability,” *Prim. Care Companion J. Clin. Psychiatry*, 2001, doi: 10.4088/pcc.v03n0105.

[54] Q. Wu, A. F. Bencaz, J. G. Hentz, and M. D. Crowell, “Selective serotonin reuptake inhibitor treatment and risk of fractures: Ameta-analysis of cohort and case-control studies,” *Osteoporos. Int.*, 2012, doi: 10.1007/s00198-011-1778-8.

[55] A. S. Bahrick, “Persistence of Sexual Dysfunction Side Effects after Discontinuation of Antidepressant Medications: Emerging Evidence,” *Open Psychol. J.*, 2008, doi: 10.2174/1874350100801010042.

[56] L. P. Koliscak and E. H. Makela, “Selective serotonin reuptake inhibitor-induced akathisia,” *J. Am. Pharm. Assoc.*, 2009, doi: 10.1331/JAPhA.2009.08083.

[57] R. M. Weinrieb, M. Auriacombe, K. G. Lynch, and J. D. Lewis, “Selective serotonin re-uptake inhibitors and the risk of bleeding,” *Expert Opinion on Drug Safety*. 2005, doi: 10.1517/14740338.4.2.337.

[58] M. Kumar, R. Bajpai, A. R. Shaik, S. Srivastava, and D. Vohora, “Alliance

between selective serotonin reuptake inhibitors and fracture risk: an updated systematic review and meta-analysis," *Eur. J. Clin. Pharmacol.*, 2020, doi: 10.1007/s00228-020-02893-1.

[59] M. P. Hengartner, S. Amendola, J. A. Kaminski, S. Kindler, T. Bschor, and M. Plöderl, "Suicide risk with selective serotonin reuptake inhibitors and new-generation serotonergic-noradrenergic antidepressants in adults: A systematic review and meta-analysis of observational studies," *medRxiv*. 2020, doi: 10.1101/2020.05.11.20098178.

[60] I. M. Anderson, P. M. Haddad, and J. Scott, "Bipolar disorder," *BMJ (Online)*. 2013, doi: 10.1136/bmj.e8508.

[61] M. M. Kurtz and R. T. Gerraty, "A Meta-Analytic Investigation of Neurocognitive Deficits in Bipolar Illness: Profile and Effects of Clinical State," *Neuropsychology*, 2009, doi: 10.1037/a0016277.

[62] D. J. Martino, C. Samamé, A. Ibañez, and S. A. Strejilevich, "Neurocognitive functioning in the premorbid stage and in the first episode of bipolar disorder: A systematic review," *Psychiatry Research*. 2015, doi: 10.1016/j.psychres.2014.12.044.

[63] C. Daban *et al.*, "Specificity of cognitive deficits in bipolar disorder versus schizophrenia: A systematic review," *Psychotherapy and Psychosomatics*. 2006, doi: 10.1159/000090891.

[64] B. Etain, C. Henry, F. Bellivier, F. Mathieu, and M. Leboyer, "Beyond genetics: Childhood affective trauma in bipolar disorder," *Bipolar Disorders*. 2008, doi: 10.1111/j.1399-5618.2008.00635.x.

[65] K. R. Merikangas *et al.*, “Prevalence and correlates of bipolar spectrum disorder in the World Mental Health Survey Initiative,” *Arch. Gen. Psychiatry*, 2011, doi: 10.1001/archgenpsychiatry.2011.12.

[66] M. Nordentoft, P. B. Mortensen, and C. B. Pedersen, “Absolute risk of suicide after first hospital contact in mental disorder,” *Arch. Gen. Psychiatry*, 2011, doi: 10.1001/archgenpsychiatry.2011.113.

[67] D. M. Novick, H. A. Swartz, and E. Frank, “Suicide attempts in bipolar I and bipolar II disorder: A review and meta-analysis of the evidence,” *Bipolar Disorders*. 2010, doi: 10.1111/j.1399-5618.2009.00786.x.

[68] M. Sajatovic, “Treatment Adherence With Lithium and Anticonvulsant Medications Among Patients With Bipolar Disorder,” *Psychiatr. Serv.*, 2007, doi: 10.1176/appi.ps.58.6.855.

[69] E. Vieta *et al.*, “Treatment options for bipolar depression: A systematic review of randomized, controlled trials,” *Journal of Clinical Psychopharmacology*. 2010, doi: 10.1097/JCP.0b013e3181f15849.

[70] M. F. Green, “Cognitive impairment and functional outcome in schizophrenia and bipolar disorder.,” *J. Clin. Psychiatry*, 2006, doi: 10.4088/jcp.1006e12.

[71] B. Bortolato, K. W. Miskowiak, C. A. Köhler, E. Vieta, and A. F. Carvalho, “Cognitive dysfunction in bipolar disorder and schizophrenia: A systematic review of meta-analyses,” *Neuropsychiatric Disease and Treatment*. 2015, doi: 10.2147/NDT.S76700.

[72] N. D. Volkow, “Substance use disorders in Schizophrenia - Clinical implications of comorbidity,” *Schizophrenia Bulletin*. 2009, doi: 10.1093/schbul/sbp016.

[73] P. F. Sullivan *et al.*, “Family history of schizophrenia and bipolar disorder as risk factors for autism,” *Arch. Gen. Psychiatry*, 2012, doi: 10.1001/archgenpsychiatry.2012.730.

[74] J. F. López-Giménez and J. González-Maeso, “Hallucinogens and serotonin 5-HT2A receptor-mediated signaling pathways,” in *Current Topics in Behavioral Neurosciences*, 2018.

[75] J. Lally and J. H. MacCabe, “Antipsychotic medication in schizophrenia: A review,” *Br. Med. Bull.*, 2015, doi: 10.1093/bmb/ldv017.

[76] K. Ehrlich *et al.*, “Dopamine D2, D3, and D4 selective phenylpiperazines as molecular probes to explore the origins of subtype specific receptor binding,” *J. Med. Chem.*, 2009, doi: 10.1021/jm900690y.

[77] L. López *et al.*, “Synthesis, 3D-QSAR, and structural modeling of benzolactam derivatives with binding affinity for the D2and D3 receptors,” *ChemMedChem*, 2010, doi: 10.1002/cmdc.201000101.

[78] M. Bantscheff, A. Scholten, and A. J. R. Heck, “Revealing promiscuous drug-target interactions by chemical proteomics,” *Drug Discovery Today*. 2009, doi: 10.1016/j.drudis.2009.07.001.

[79] J. Scheiber *et al.*, “Mapping adverse drug reactions in chemical space,” *J. Med. Chem.*, 2009, doi: 10.1021/jm801546k.

[80] V. Aderhold, S. Weinmann, C. Hägele, and A. Heinz, “Frontal brain volume reduction due to antipsychotic drugs?,” *Nervenarzt*, 2015, doi: 10.1007/s00115-014-4027-5.

[81] C. H. Rojas-Fernandez, “Pharmacodynamics,” in *Applied Clinical*

Pharmacokinetics and Pharmacodynamics of Psychopharmacological Agents, 2016.

[82] T. Schlick, "Molecular Dynamics: Basics," 2010.

[83] A. Heinecke, W. Eckhardt, M. Horsch, and H. J. Bungartz, "Molecular dynamics simulation," in *SpringerBriefs in Computer Science*, 2015.

[84] Y. Lee, S. Basith, and S. Choi, "Recent Advances in Structure-Based Drug Design Targeting Class A G Protein-Coupled Receptors Utilizing Crystal Structures and Computational Simulations," *Journal of Medicinal Chemistry*. 2018, doi: 10.1021/acs.jmedchem.6b01453.

[85] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, 1953, doi: 10.1063/1.1699114.

[86] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method: Third Edition*. 2016.

[87] I. Bahar, B. Erman, T. Haliloglu, and R. L. Jernigan, "Efficient characterization of collective motions and interresidue correlations in proteins by low-resolution simulations," *Biochemistry*, 1997, doi: 10.1021/bi971611f.

[88] S. Monti, "Monte Carlo dynamics and molecular dynamics simulations in polymer science," *Phys. Chem. Chem. Phys.*, 1995, doi: 10.1039/c0cp02898c.

[89] A. Zen, Y. Luo, G. Mazzola, L. Guidoni, and S. Sorella, "Ab initio molecular dynamics simulation of liquid water by quantum Monte Carlo," *J. Chem. Phys.*, 2015, doi: 10.1063/1.4917171.

[90] J. A. McCammon and S. C. Harvey, *Dynamics of Proteins and Nucleic Acids*.

1987.

[91] I. A. Guedes, C. S. de Magalhães, and L. E. Dardenne, “Receptor-ligand molecular docking,” *Biophysical Reviews*. 2014, doi: 10.1007/s12551-013-0130-2.

[92] X. Yuan and Y. Xu, “Recent trends and applications of molecular modeling in GPCR–Ligand recognition and structure-based drug design,” *International Journal of Molecular Sciences*. 2018, doi: 10.3390/ijms19072105.

[93] A. E. Cardenas and R. Elber, “Markovian and Non-Markovian Modeling of Membrane Dynamics with Milestoning,” *J. Phys. Chem. B*, 2016, doi: 10.1021/acs.jpcb.6b01890.

[94] R. Elber, A. Fathizadeh, P. Ma, and H. Wang, “Modeling molecular kinetics with Milestoning,” *WIREs Comput. Mol. Sci.*, vol. n/a, no. n/a, p. e1512, Dec. 2020, doi: https://doi.org/10.1002/wcms.1512.

[95] Y. Shi *et al.*, “Polarizable atomic multipole-based AMOEBA force field for proteins,” *J. Chem. Theory Comput.*, 2013, doi: 10.1021/ct4003702.

[96] M. S. Miller, W. K. Lay, and A. H. Elcock, “Osmotic Pressure Simulations of Amino Acids and Peptides Highlight Potential Routes to Protein Force Field Parameterization,” *J. Phys. Chem. B*, 2016, doi: 10.1021/acs.jpcb.6b01902.

[97] R. Elber, “Perspective: Computer simulations of long time dynamics,” *J. Chem. Phys.*, vol. 144, no. 6, p. 60901, Feb. 2016, doi: 10.1063/1.4940794.

[98] J. Hochuli, A. Helbling, T. Skaist, M. Ragoza, and D. R. Koes, “Visualizing convolutional neural network protein-ligand scoring,” *J. Mol. Graph. Model.*, 2018, doi: 10.1016/j.jmgm.2018.06.005.

[99] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*. 2015, doi:

10.1038/nature14539.

[100] E. Ghosh, P. Kumari, D. Jaiman, and A. K. Shukla, “Methodological advances: The unsung heroes of the GPCR structural revolution,” *Nature Reviews Molecular Cell Biology*. 2015, doi: 10.1038/nrm3933.

[101] T. M. Mitchell, “Machine learning. 1997,” *Burr Ridge, McGraw Hill*, 1997.

[102] A. C. Ian Goodfellow, Yoshua Bengio, “Deep Learning Book,” *Deep Learn.*, 2015.

[103] A. Zheng, *Evaluating Machine Learning Models A Beginner’s Guide to Key Concepts and Pitfalls*. 2015.

[104] N. Chinchor, “MUC-4 Evaluation Metrics,” in *Proc. of the Fourth Message Understanding Conference*, 1992, pp. 22–29.

[105] P. Bruce and A. Bruce, *Practical Statistics for Data Scientists*. 2017.

[106] C. J. Willmott and K. Matsuura, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance,” *Clim. Res.*, 2005, doi: 10.3354/cr030079.

[107] T. Gneiting and P. Vogel, “Receiver Operating Characteristic (ROC) curves,” *arXiv*. 2018, doi: 10.1007/978-1-4419-9863-7_242.

[108] K. Boyd, K. H. Eng, and C. D. Page, “Area under the precision-recall curve: Point estimates and confidence intervals,” 2013, doi: 10.1007/978-3-642-40994-3_29.

[109] J. Davis and M. Goadrich, “The relationship between precision-recall and ROC curves,” 2006, doi: 10.1145/1143844.1143874.

[110] L. A. Jeni, J. F. Cohn, and F. De La Torre, “Facing imbalanced data - Recommendations for the use of performance metrics,” 2013, doi:

10.1109/ACII.2013.47.

[111] E. Bisong and E. Bisong, “Regularization for Deep Learning,” in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, 2019.

[112] Š. Raudys and A. K. Jain, “Small sample size problems in designing artificial neural networks,” in *Machine Intelligence and Pattern Recognition*, 1991.

[113] I. De Feis, “Dimensionality reduction,” in *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 2018.

[114] X. Huang, L. Wu, and Y. Ye, “A Review on Dimensionality Reduction Techniques,” *Int. J. Pattern Recognit. Artif. Intell.*, 2019, doi: 10.1142/S0218001419500174.

[115] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, 2011.

[116] I. Iguyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*. 2003, doi: 10.1162/153244303322753616.

[117] L. C. Lee, C. Y. Liong, and A. A. Jemain, “Partial least squares-discriminant analysis (PLS-DA) for classification of high-dimensional (HD) data: A review of contemporary practice strategies and knowledge gaps,” *Analyst*. 2018, doi: 10.1039/c8an00599k.

[118] H. Wold, “Estimation of principal components and related models by iterative least squares,” in *Multivariate Analysis*, 1966.

[119] H. WOLD, “Nonlinear Iterative Partial Least Squares (NIPALS) Modelling: Some Current Developments,” in *Multivariate Analysis–III*, 1973.

[120] I. T. Jollife and J. Cadima, “Principal component analysis: A review and recent developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2016, doi: 10.1098/rsta.2015.0202.

[121] R. Sjögren, K. Stridh, T. Skotare, and J. Trygg, “Multivariate patent analysis—Using chemometrics to analyze collections of chemical and pharmaceutical patents,” *J. Chemom.*, 2020, doi: 10.1002/cem.3041.

[122] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” *J. Educ. Psychol.*, 1933, doi: 10.1037/h0071325.

[123] G. Shobha and S. Rangaswamy, “Machine Learning,” in *Handbook of Statistics*, 2018.

[124] J. Trygg and S. Wold, “Orthogonal projections to latent structures (O-PLS),” *J. Chemom.*, 2002, doi: 10.1002/cem.695.

[125] T. K. Ho, “Random decision forests,” 1995, doi: 10.1109/ICDAR.1995.598994.

[126] L. Breiman, “Random forests,” *Mach. Learn.*, 2001, doi: 10.1023/A:1010933404324.

[127] R. Rojas, “AdaBoost and the Super Bowl of Classifiers A Tutorial Introduction to Adaptive Boosting,” *Writing*, 2009.

[128] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” *Annals of Statistics*. 2000, doi: 10.1214/aos/1016218223.

[129] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, 1995, doi: 10.1023/A:1022627411411.

[130] S. Developers, “Support Vector Machines — scikit-learn 0.17.1 documentation,”

http://scikit-learn.org, 2014. .

[131] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychol. Rev.*, 1958, doi: 10.1037/h0042519.

[132] M. Minsky and S. Papert, “Perceptron: an introduction to computational geometry,” *MIT Press. Cambridge, Expand. Ed.*, 1969.

[133] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization methods for large-scale machine learning,” *SIAM Review*. 2018, doi: 10.1137/16M1080173.

[134] L. Bottou, “Online Algorithms and Stochastic Approximation,” *Online Learning and Neural Networks*. 1998.

[135] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, 1986, doi: 10.1007/bf00116251.

[136] W. A. Belson, “Matching and Prediction on the Principle of Biological Classification,” *Appl. Stat.*, 1959, doi: 10.2307/2985543.

[137] Scikit-Learn Developers, “3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.18.1 documentation,” *Scikit-learn documentation*, 2016. .

[138] C. Cao *et al.*, “Deep Learning and Its Applications in Biomedicine,” *Genomics, Proteomics and Bioinformatics*. 2018, doi: 10.1016/j.gpb.2017.07.003.

[139] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer, “An Introductory Review of Deep Learning for Prediction Models With Big Data,” *Front. Artif. Intell.*, 2020, doi: 10.3389/frai.2020.00004.

[140] Scikit-Learn Developers, “Neural network models (supervised),” *Scikit-Learn Documentation*, 2018. .

[141] Y. A. LeCun, L. Bottou, G. B. Orr, and K. R. Müller, “Efficient backprop,” *Lect.*

Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2012, doi: 10.1007/978-3-642-35289-8_3.

[142] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, 1989, doi: 10.1162/neco.1989.1.4.541.

[143] P. J. Werbos, "The roots of backpropagation: from ordered derivatives to neural networks ... - Paul John Werbos - Google Libros," *Doctoral thesis*, 1974. .

[144] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, 1980, doi: 10.1007/BF00344251.

[145] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognit.*, 1982, doi: 10.1016/0031-3203(82)90024-3.

[146] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, 1988, doi: 10.1016/0893-6080(88)90014-7.

[147] K. Fukushima, "Neocognitron," *Scholarpedia*, 2007, doi: 10.4249/scholarpedia.1717.

[148] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," 2010, doi: 10.1007/978-3-642-15825-4_10.

[149] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "2012 AlexNet," *Adv. Neural Inf. Process. Syst.*, 2012, doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

[150] K. Yamaguchi, K. Sakamoto, T. Akabane, and Y. Fujimoto, "A neural network for speaker-independent isolated word recognition," in *ICSLP*, 1990, pp. 1077–1080.

[151] J. J. Weng, N. Ahuja, and T. S. Huang, “Learning recognition and segmentation of 3-D objects from 2-D images,” 1993, doi: 10.1109/iccv.1993.378228.

[152] Q. N. Vo, P. Mahinthichaichan, J. Shen, and C. R. Ellis, “How μ-opioid receptor recognizes fentanyl,” *bioRxiv*. 2020, doi: 10.1101/2020.08.16.253013.

[153] A. Gandhimathi and R. Sowdhamini, “Molecular modelling of human 5-hydroxytryptamine receptor (5-HT2A) and virtual screening studies towards the identification of agonist and antagonist molecules,” *J. Biomol. Struct. Dyn.*, 2016, doi: 10.1080/07391102.2015.1062802.

[154] X. Du, S. Sun, C. Hu, Y. Yao, Y. Yan, and Y. Zhang, “DeepPPI: Boosting Prediction of Protein-Protein Interactions with Deep Neural Networks,” *J. Chem. Inf. Model.*, 2017, doi: 10.1021/acs.jcim.7b00028.

[155] Q. Shi *et al.*, “DNN-Dom: Predicting protein domain boundary from sequence alone by deep neural network,” *Bioinformatics*, 2019, doi: 10.1093/bioinformatics/btz464.

[156] C. Nantasenamat, C. Isarankura-Na-Ayudhya, T. Naenna, and V. Prachayasittikul, “A practical overview of quantitative structure-activity relationship,” *EXCLI Journal*. 2009.

[157] Y. G. Chushak, H. W. Shows, J. M. Gearhart, and H. A. Pangburn, “In silico identification of protein targets for chemical neurotoxins using ToxCast in vitro data and read-across within the QSAR toolbox,” *Toxicol. Res. (Camb).*, 2018, doi: 10.1039/c7tx00268h.

[158] D. J. Dix, K. A. Houck, M. T. Martin, A. M. Richard, R. W. Setzer, and R. J. Kavlock, “The toxcast program for prioritizing toxicity testing of environmental

chemicals," *Toxicol. Sci.*, 2007, doi: 10.1093/toxsci/kfl103.

[159] J. Liu *et al.*, "Predicting hepatotoxicity using ToxCast in vitro bioactivity and chemical structure," *Chem. Res. Toxicol.*, 2015, doi: 10.1021/tx500501h.

[160] A. M. Richard *et al.*, "ToxCast Chemical Landscape: Paving the Road to 21st Century Toxicology," *Chemical Research in Toxicology*. 2016, doi: 10.1021/acs.chemrestox.6b00135.

[161] M. Silva *et al.*, "A Comparison of ToxCast Test Results with In Vivo and Other In Vitro Endpoints for Neuro, Endocrine, and Developmental Toxicities: A Case Study Using Endosulfan and Methidathion," *Birth Defects Res. Part B - Dev. Reprod. Toxicol.*, 2015, doi: 10.1002/bdrb.21140.

[162] V. Isberg *et al.*, "Generic GPCR residue numbers - Aligning topology maps while minding the gaps," *Trends in Pharmacological Sciences*. 2015, doi: 10.1016/j.tips.2014.11.001.

[163] I. V. Tetko *et al.*, "Virtual computational chemistry laboratory - Design and description," *J. Comput. Aided. Mol. Des.*, 2005, doi: 10.1007/s10822-005-8694-y.

[164] I. Bahar, T. R. Lezon, L. W. Yang, and E. Eyal, "Global dynamics of proteins: Bridging between structure and function," *Annual Review of Biophysics*. 2010, doi: 10.1146/annurev.biophys.093008.131258.

[165] I. D. Brown, "Topology and chemistry," *Structural Chemistry*. 2002, doi: 10.1023/A:1015872125545.

[166] K. A. Jacobson, S. Costanzi, and S. Paoletta, "Computational studies to predict or explain G protein coupled receptor polypharmacology," *Trends in Pharmacological Sciences*. 2014, doi: 10.1016/j.tips.2014.10.009.

[167] S. Basith *et al.*, “Exploring G protein-coupled receptors (GPCRs) ligand space via cheminformatics approaches: Impact on rational drug design,” *Frontiers in Pharmacology*. 2018, doi: 10.3389/fphar.2018.00128.

[168] G. M. Hu, T. L. Mai, and C. M. Chen, “Visualizing the GPCR Network: Classification and Evolution,” *Sci. Rep.*, 2017, doi: 10.1038/s41598-017-15707-9.

[169] S. Raschka and B. Kaufman, “Machine learning and AI-based approaches for bioactive ligand discovery and GPCR-ligand recognition,” *Methods*, vol. 180, pp. 89–110, 2020, doi: https://doi.org/10.1016/j.ymeth.2020.06.016.

[170] M. Ahmed, H. J. Hasani, S. Kalyaanamoorthy, and K. Barakat, “GPCR_LigandClassify.py; a rigorous machine learning classifier for GPCR targeting compounds,” *Sci. Rep.*, vol. 11, no. 1, p. 9510, 2021, doi: 10.1038/s41598-021-88939-5.

[171] Z. R. Li, H. H. Lin, L. Y. Han, L. Jiang, X. Chen, and Y. Z. Chen, “PROFEAT: A web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence,” *Nucleic Acids Res.*, 2006, doi: 10.1093/nar/gkl305.

[172] C. W. Yap, “PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints,” *J. Comput. Chem.*, 2011, doi: 10.1002/jcc.21707.

[173] N. M. O’Boyle, C. Morley, and G. R. Hutchison, “Pybel: A Python wrapper for the OpenBabel cheminformatics toolkit,” *Chem. Cent. J.*, 2008, doi: 10.1186/1752-153X-2-5.

[174] M. E. Cooper, “Chemoinformatics: Concepts, Methods and Tools for Drug Discovery,” *Drug Discov. Today*, 2004, doi: 10.1016/s1359-6446(04)03262-3.

[175] R. Todeschini and V. Consonni, *Handbook of Molecular Descriptors*. 2000.

[176] H. Hong *et al.*, "Mold2, molecular descriptors from 2D structures for chemoinformatics and toxicoinformatics," *J. Chem. Inf. Model.*, 2008, doi: 10.1021/ci800038f.

[177] A. R. Leach and V. J. Gillet, "Molecular Descriptors," in *An Introduction To Chemoinformatics*, 2007.

[178] O. C. Redfern, B. Dessailly, and C. A. Orengo, "Exploring the structure and function paradigm," *Current Opinion in Structural Biology*. 2008, doi: 10.1016/j.sbi.2008.05.007.

[179] D. R. Flower, A. C. T. North, and C. E. Sansom, "The lipocalin protein family: Structural and sequence overview," *Biochimica et Biophysica Acta - Protein Structure and Molecular Enzymology*. 2000, doi: 10.1016/S0167-4838(00)00148-5.

[180] N. Nagano, C. A. Orengo, and J. M. Thornton, "One fold with many functions: The evolutionary relationships between TIM barrel families based on their sequences, structures and functions," *Journal of Molecular Biology*. 2002, doi: 10.1016/S0022-2836(02)00649-6.

[181] K. Illergård, D. H. Ardell, and A. Elofsson, "Structure is three to ten times more conserved than sequence - A study of structural response in protein cores," *Proteins Struct. Funct. Bioinforma.*, 2009, doi: 10.1002/prot.22458.

[182] I. Bahar, A. R. Atilgan, and B. Erman, "Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential," *Fold. Des.*, 1997, doi: 10.1016/S1359-0278(97)00024-2.

[183] A. R. Atilgan, S. R. Durell, R. L. Jernigan, M. C. Demirel, O. Keskin, and I. Bahar, “Anisotropy of fluctuation dynamics of proteins with an elastic network model,” *Biophys. J.*, 2001, doi: 10.1016/S0006-3495(01)76033-X.

[184] P. Doruker, A. R. Atilgan, and I. Bahar, “Dynamics of proteins predicted by molecular simulations and analytical approaches: Application to α-amylase inhibitor,” *Proteins Struct. Funct. Genet.*, 2000, doi: 10.1002/1097-0134(20000815)40:3<512::AID-PROT180>3.0.CO;2-M.

[185] E. Eyal, L. W. Yang, and I. Bahar, “Anisotropic network model: Systematic evaluation and a new web interface,” *Bioinformatics*, 2006, doi: 10.1093/bioinformatics/btl448.

[186] T. Haliloglu, I. Bahar, and B. Erman, “Gaussian dynamics of folded proteins,” *Phys. Rev. Lett.*, 1997, doi: 10.1103/PhysRevLett.79.3090.

[187] R. Sarkar, “Stiffening of flexible SUMO1 protein upon peptide-binding: Analysis with anisotropic network model,” *Math. Biosci.*, 2018, doi: 10.1016/j.mbs.2017.11.008.

[188] A. Das, M. Gur, M. H. Cheng, S. Jo, I. Bahar, and B. Roux, “Exploring the Conformational Transitions of Biomolecular Systems Using a Simple Two-State Anisotropic Network Model,” *PLoS Comput. Biol.*, 2014, doi: 10.1371/journal.pcbi.1003521.

[189] K. Xia, K. Opron, and G. W. Wei, “Multiscale Gaussian network model (mGNM) and multiscale anisotropic network model (mANM),” *J. Chem. Phys.*, 2015, doi: 10.1063/1.4936132.

[190] H. Li, Y. Y. Chang, J. Y. Lee, I. Bahar, and L. W. Yang, “DynOmics: Dynamics

of structural proteome and beyond," *Nucleic Acids Res.*, 2017, doi: 10.1093/nar/gkx385.

[191] H. Wako and S. Endo, "Normal mode analysis as a method to derive protein dynamics information from the Protein Data Bank," *Biophysical Reviews*. 2017, doi: 10.1007/s12551-017-0330-2.

[192] M. Akdel, J. Durairaj, D. de Ridder, and A. D. J. van Dijk, "Caretta – A multiple protein structure alignment and feature extraction suite," *Comput. Struct. Biotechnol. J.*, 2020, doi: 10.1016/j.csbj.2020.03.011.

[193] B. Brooks and M. Karplus, "Harmonic dynamics of proteins: Normal modes and fluctuations in bovine pancreatic trypsin inhibitor," *Proc. Natl. Acad. Sci. U. S. A.*, 1983, doi: 10.1073/pnas.80.21.6571.

[194] Q. Cui and I. Bahar, *Normal mode analysis: Theory and applications to biological and chemical systems*. 2005.

[195] C. Xu, D. Tobi, and I. Bahar, "Allosteric changes in protein structure computed by a simple mechanical model: Hemoglobin T ↔ R2 transition," *J. Mol. Biol.*, 2003, doi: 10.1016/j.jmb.2003.08.027.

[196] I. Bahar, A. R. Atilgan, M. C. Demirel, and B. Erman, "Vibrational dynamics of folded proteins: Significance of slow and fast motions in relation to function and stability," *Phys. Rev. Lett.*, 1998, doi: 10.1103/PhysRevLett.80.2733.

[197] M. Delarue and Y. H. Sanejouand, "Simplified normal mode analysis of conformational transitions in DNA-dependent polymerases: The Elastic Network Model," *J. Mol. Biol.*, 2002, doi: 10.1016/S0022-2836(02)00562-4.

[198] S. Majumder, D. Chaudhuri, J. Datta, and K. Giri, "Exploring the intrinsic

dynamics of SARS-CoV-2, SARS-CoV and MERS-CoV spike glycoprotein through normal mode analysis using anisotropic network model," *J. Mol. Graph. Model.*, 2021, doi: 10.1016/j.jmgm.2020.107778.

[199] D. Kolan, G. Fonar, and A. O. Samson, "Elastic network normal mode dynamics reveal the GPCR activation mechanism," *Proteins Struct. Funct. Bioinforma.*, 2014, doi: 10.1002/prot.24426.

[200] I. G. Tikhonova, R. B. Best, S. Engel, M. C. Gershengorn, G. Hummer, and S. Costanzi, "Atomistic insights into rhodopsin activation from a dynamic model," *J. Am. Chem. Soc.*, 2008, doi: 10.1021/ja0765520.

[201] I. Kufareva and R. Abagyan, "Methods of protein structure comparison," *Methods Mol. Biol.*, 2012, doi: 10.1007/978-1-61779-588-6_10.

[202] M. Stamm and L. R. Forrest, "Structure alignment of membrane proteins: Accuracy of available tools and a consensus strategy," *Proteins Struct. Funct. Bioinforma.*, 2015, doi: 10.1002/prot.24857.

[203] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jul. 2002, doi: 10.1613/jair.953.

[204] H. M. Berman *et al.*, "The Protein Data Bank," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, Jan. 2000, doi: 10.1093/nar/28.1.235.

[205] G. Pándy-Szekeres *et al.*, "GPCRdb in 2018: Adding GPCR structure models and ligands," *Nucleic Acids Res.*, 2018, doi: 10.1093/nar/gkx1109.

[206] C. Munk *et al.*, "An online resource for GPCR structure determination and analysis," *Nat. Methods*, 2019, doi: 10.1038/s41592-018-0302-x.

[207] D. S. Wishart *et al.*, “DrugBank: a comprehensive resource for in silico drug discovery and exploration.,” *Nucleic Acids Res.*, vol. 34, no. Database issue, pp. D668-72, Jan. 2006, doi: 10.1093/nar/gkj067.

[208] “MATLAB. (2010). version 7.10.0 (R2010a). Natick, Massachusetts: The MathWorks Inc.” .

[209] G. Landrum, “Getting Started with the RDKit in Python — The RDKit 2020.03.1 documentation,” *RDKit*, 2016. .

[210] D. Elreedy and A. F. Atiya, “A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance,” *Inf. Sci. (Ny).*, vol. 505, pp. 32–64, 2019, doi: https://doi.org/10.1016/j.ins.2019.07.070.

[211] W. Kabsch and C. Sander, “Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features,” *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983, doi: 10.1002/bip.360221211.

[212] “Jmol: an open-source Java viewer for chemical structures in 3D. http://www.jmol.org/.” .

[213] E. F. Pettersen *et al.*, “UCSF ChimeraX: Structure visualization for researchers, educators, and developers,” *Protein Sci.*, 2021, doi: 10.1002/pro.3943.

[214] C. Vega, J. L. F. Abascal, and I. Nezbeda, “Vapor-liquid equilibria from the triple point up to the critical point for the new generation of TIP4P-like models: TIP4P/Ew, TIP4P/2005, and TIP4P/ice,” *J. Chem. Phys.*, vol. 125, no. 3, p. 34503, Jul. 2006, doi: 10.1063/1.2215612.

[215] J. E. J. Mills and P. M. Dean, “Three-dimensional hydrogen-bond geometry and probability information from a crystal survey,” *J. Comput. Aided. Mol. Des.*, vol.

10, no. 6, pp. 607–622, 1996, doi: 10.1007/BF00134183.

[216] M. D. Intriligator, “Drug Evaluations: Type I vs. Type II Errors.,” *UCLA Res. Progr. Pharm. Econ. Policy.*, 1996, [Online]. Available: https://escholarship.org/uc/item/5fg9n284.

[217] PyTorch, “PyTorch documentation - PyTorch master documentation,” *PyTorch*, 2019. .

[218] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *J. Mol. Biol.*, vol. 48, no. 3, pp. 443–453, 1970, doi: https://doi.org/10.1016/0022-2836(70)90057-4.

[219] H. Frauenfelder, S. G. Sligar, and P. G. Wolynes, “The energy landscapes and motions of proteins,” *Science (80-.).*, 1991, doi: 10.1126/science.1749933.

[220] M.-H. Seo, J. Park, E. Kim, S. Hohng, and H.-S. Kim, “Protein conformational dynamics dictate the binding affinity for a ligand,” *Nat. Commun.*, vol. 5, no. 1, p. 3724, 2014, doi: 10.1038/ncomms4724.

[221] Y. Kang *et al.*, “Cryo-EM structure of human rhodopsin bound to an inhibitory G protein,” *Nature*, vol. 558, no. 7711, pp. 553–558, 2018, doi: 10.1038/s41586-018-0215-y.

[222] B. Trzaskowski, D. Latek, S. Yuan, U. Ghoshdastider, A. Debinski, and S. Filipek, “Action of molecular switches in GPCRs--theoretical and experimental studies,” *Curr. Med. Chem.*, vol. 19, no. 8, pp. 1090–1109, 2012, doi: 10.2174/092986712799320556.

[223] J. A. Coleman and E. Gouaux, “Structural basis for recognition of diverse antidepressants by the human serotonin transporter,” *Nat. Struct. Mol. Biol.*, 2018,

doi: 10.1038/s41594-018-0026-8.

[224] W. Yin *et al.*, “Crystal structure of the human 5-HT1B serotonin receptor bound to an inverse agonist,” *Cell Discov.*, vol. 4, no. 1, p. 12, 2018, doi: 10.1038/s41421-018-0009-2.

[225] N. Almaula, B. J. Ebersole, D. Zhang, H. Weinstein, and S. C. Sealfon, “Mapping the binding site pocket of the serotonin 5-Hydroxytryptamine2A receptor. Ser3.36(159) provides a second interaction site for the protonated amine of serotonin but not of lysergic acid diethylamide or bufotenin.,” *J. Biol. Chem.*, vol. 271, no. 25, pp. 14672–14675, Jun. 1996, doi: 10.1074/jbc.271.25.14672.

[226] A. Jabeen and S. Ranganathan, “Applications of machine learning in GPCR bioactive ligand discovery,” *Curr. Opin. Struct. Biol.*, vol. 55, pp. 66–76, 2019, doi: 10.1016/j.sbi.2019.03.022.

[227] S. H. Lee, K.-Y. Kim, and Y. Shin, “Effective Feature Selection Method for Deep Learning-Based Automatic Modulation Classification Scheme Using Higher-Order Statistics,” *Applied Sciences* , vol. 10, no. 2. 2020, doi: 10.3390/app10020588.

[228] S. Zhang *et al.*, “A deep learning framework for modeling structural features of RNA-binding protein targets,” *Nucleic Acids Res.*, 2015, doi: 10.1093/nar/gkv1025.

[229] D. Luethi and M. E. Liechti, “Designer drugs: mechanism of action and adverse effects,” *Arch. Toxicol.*, vol. 94, no. 4, pp. 1085–1133, Apr. 2020, doi: 10.1007/s00204-020-02693-7.

[230] L. R. Gagnon, C. Sadasivan, K. Perera, and G. Y. Oudit, “Cardiac Complications of Common Drugs of Abuse: Pharmacology, Toxicology and Management.,” *Can.*

J. Cardiol., Nov. 2021, doi: 10.1016/j.cjca.2021.10.008.

[231] A.-N. Samaha, S. Y.-S. Khoo, C. R. Ferrario, and T. E. Robinson, "Dopamine 'ups and downs' in addiction revisited.," *Trends Neurosci.*, vol. 44, no. 7, pp. 516–526, Jul. 2021, doi: 10.1016/j.tins.2021.03.003.

[232] F. Boeckler, H. Lanig, and P. Gmeiner, "Modeling the similarity and divergence of dopamine D2-like receptors and identification of validated ligand-receptor complexes," *J. Med. Chem.*, 2005, doi: 10.1021/jm049612a.

[233] I. Z. Ben Zion *et al.*, "Polymorphisms in the dopamine D4 receptor gene (DRD4) contribute to individual differences in human sexual behavior: desire, arousal and sexual function," *Mol. Psychiatry*, vol. 11, no. 8, pp. 782–786, 2006, doi: 10.1038/sj.mp.4001832.

[234] G. Sullivan and D. Lukoff, "Sexual side effects of antipsychotic medication: evaluation and interventions.," *Hosp. Community Psychiatry*, vol. 41, no. 11, pp. 1238–1241, Nov. 1990, doi: 10.1176/ps.41.11.1238.

[235] K. J. Allahdadi, R. C. A. Tostes, and R. C. Webb, "Female sexual dysfunction: therapeutic options and experimental challenges," *Cardiovasc. Hematol. Agents Med. Chem.*, vol. 7, no. 4, pp. 260–269, Oct. 2009, doi: 10.2174/187152509789541882.

[236] U. Simonsen, S. Comerma-Steffensen, and K.-E. Andersson, "Modulation of Dopaminergic Pathways to Treat Erectile Dysfunction," *Basic Clin. Pharmacol. Toxicol.*, vol. 119, no. S3, pp. 63–74, Oct. 2016, doi: https://doi.org/10.1111/bcpt.12653.

[237] J. D. Brioni *et al.*, "Activation of dopamine D4 receptors by ABT-724 induces

penile erection in rats," *Proc. Natl. Acad. Sci. U. S. A.*, 2004, doi: 10.1073/pnas.0308292101.

[238] J. D. Brioni and R. B. Moreland, "Dopamine D4 receptors and the regulation of penile erection," *Drug Discovery Today: Therapeutic Strategies*. 2006, doi: 10.1016/j.ddstr.2006.10.006.

[239] K. J. Burghardt, A. S. Khoury, Z. Msallaty, Z. Yi, and B. Seyoum, "Antipsychotic Medications and DNA Methylation in Schizophrenia and Bipolar Disorder: A Systematic Review," *Pharmacotherapy*, vol. 40, no. 4, pp. 331–342, 2020, doi: 10.1002/phar.2375.

[240] B.-C. Ho, N. C. Andreasen, S. Ziebell, R. Pierson, and V. Magnotta, "Long-term Antipsychotic Treatment and Brain Volumes," *Arch. Gen. Psychiatry*, 2011, doi: 10.1001/archgenpsychiatry.2010.199.

[241] S. Chopra *et al.*, "Differentiating the effect of antipsychotic medication and illness on brain volume reductions in first-episode psychosis: A Longitudinal, Randomised, Triple-blind, Placebo-controlled MRI Study," *Neuropsychopharmacology*, 2021, doi: 10.1038/s41386-021-00980-0.

[242] R. D. Hipp, "SQLite." 2020, [Online]. Available: https://www.sqlite.org/index.html.

[243] S. Imambi, K. B. Prakash, and G. R. Kanagachidambaresan, "PyTorch," 2021.

[244] E. Eyal, G. Lum, and I. Bahar, "The anisotropic network model web server at 2015 (ANM 2.0)," *Bioinformatics*, 2015, doi: 10.1093/bioinformatics/btu847.

Appendix A. Receptor and Drug Features

A.1 Receptor feature engineering from ANM output

Receptor features each correspond to a given residue in which residue numbering corresponds to the multiple structural alignment for the given dataset. Following calculation of atomic fluctuations as described in **3.4**, each protein is encoded by a feature vector in which index corresponds to residue number and elements correspond to mean atomic fluctuations for the alpha carbon of the given residue position for use in machine learning classifiers. An example representation of input data format is given in **Table A1.1**. All ANM output can be found in a Bioinformatics Research Group Github Repository.

	AlignedRes_1	AlignedRes_2	AlignedRes_3	…	AlignedRes_*n*
GPCR_1	Cα_fluct_1$_{GPCR_1}$	Cα_fluct_2$_{GPCR_1}$	Cα_fluct_3$_{GPCR_1}$	…	Cα_fluct_n_{GPCR_1}
GPCR_2	Cα_fluct_1$_{GPCR_2}$	Cα_fluct_2$_{GPCR_2}$	Cα_fluct_3$_{GPCR_2}$	…	Cα_fluct_n_{GPCR_2}
GPCR_3	Cα_fluct_1$_{GPCR_3}$	Cα_fluct_2$_{GPCR_3}$	Cα_fluct_3$_{GPCR_3}$	…	Cα_fluct_n_{GPCR_3}
…	…	…	…	…	…
GPCR_*n*	Cα_fluct_1$_{GPCR_n}$	Cα_fluct_2$_{GPCR_n}$	Cα_fluct_3$_{GPCR_n}$	…	Cα_fluct_n_{GPCR_n}

Table A1.1 Format of data features for protein receptors. Each protein corresponds to a data observation in which each feature describes the mean atomic fluctuations for each aligned residue as determined by an anisotropic network model.

A.2 Drug feature encoding

Chemical features were generated from SMILES using the RDKit Chem.Features package in Python 3.6. [209] Features belong to the following molecular descriptor categories:

Gasteiger/Marsili Partial Charges	**BalabanJ**	**BertzCT**
Ipc	HallKierAlpha	Kappa1-Kappa3
Phi	Chi0,Chi1	Chi0n-Chi4n
Chi0v	Chi4v	MolLogP
MolMR	MolWt	ExactMolWt
ExactMolWt	HeavyAtomCount	HeavyAtomMolWt
NHOHCount	NOCount	NumHAcceptors
NumHDonors	NumHeteroatoms	NumRotatableBonds
NumValenceElectrons	NumAmideBonds	Num{Aromatic,Saturated,Aliphatic}Rings
Num{Aromatic,Saturated,Aliphatic}{Hetero,Carbo}Cycles	RingCount	FractionCSP3
NumSpiroAtoms	NumBridgeheadAtoms	TPSA

LabuteASA	PEOE_VSA1 – PEOE_VSA14	SMR_VSA1- SMR_VSA10
SlogP_VSA1 – SlogP_VSA12	Estate_VSA1- Estate_VSA11	VSA_Estate1 – VSA_Estate10
MQNs	Topliss Fragments	Autocorr2D
BCUT2D	Eccentricity	SSSR

Table A2.1 Molecular descriptors selected from the RDKit package to be used as features in machine learning prediction. Expanded list with notes for each descriptor calculation as well as code can be found in the RDKit documentation. [209]

Appendix B. Code Source and Examples

B.1. Dataset generation and cleaning

To obtain comprehensive datasets of GPCR interaction activities, all datasets (GPCRdb for receptor and receptor structure data [23], PDSP for psychoactive drug activities [26], and DrugBank [207] for ligand structure data) were imported into SQLite [242] for database management and combined using the relational algebra join functions in SQLite to obtain a database containing only activities for interactions between GPCRs and psychoactive drugs in which the GPCR has an experimentally solved structure and the ligand SMILES can be obtained. Protocol and code for using SQLite join functions can be found in the SQLite documentation. [242]

Following the generation of the GPCR/psychoactive drug database, data cleaning was performed in OpenRefine to remove any duplicate observations or observations containing missing values, encode activities into a binary output, and remove interactions with multiple entries that are not in agreement as previously described in **3.2**. OpenRefine source code can be obtained at openrefine.github.com.

B.2. Data loaders

All data loaders use the Dataset and DataLoader module within PyTorch. [243]

```
## train data
class trainData(Dataset):

        def __init__(self, X_data, y_data):
                self.X_data = X_data
                self.y_data = y_data
        def __getitem__(self, index):
        return self.X_data[index], self.y_data[index]

        def __len__ (self):
                return len(self.X_data)
train_data = trainData(torch.FloatTensor(X_train),
                        torch.FloatTensor(y_train))
## test data
class testData(Dataset):

        def __init__(self, X_data):
                self.X_data = X_data

        def __getitem__(self, index):
                return self.X_data[index]

        def __len__ (self):
                return len(self.X_data)

test_data = testData(torch.FloatTensor(X_test))


train_loader = DataLoader(dataset=train_data, batch_size=BATCH_SIZE,␣
,shuffle=True)
test_loader = DataLoader(dataset=test_data, batch_size=1)
```

B.3. Deep learning classifier

Deep neural network architecture was built using the torch.nn module within PyTorch.

[243]

```
class binaryClassification(nn.Module):
        def __init__(self):
                super(binaryClassification, self).__init__().
                self.layer_1 = nn.Linear(numFeat, 1400)
                self.layer_2 = nn.Linear(1400, 1400)
                self.layer_3 = nn.Linear(1400, 700)
                self.layer_4 = nn.Linear(700,350)
                self.layer_out = nn.Linear(350, 1)

                self.relu = nn.ReLU()
                self.dropout = nn.Dropout(p=0.1)
                self.batchnorm1 = nn.BatchNorm1d(1400)
                self.batchnorm2 = nn.BatchNorm1d(1400)
                self.batchnorm3 = nn.BatchNorm1d(700)
                self.batchnorm4 = nn.BatchNorm1d(350)
                self.dropout2 = nn.Dropout(p=0.8)

        def forward(self, inputs):
                x = self.relu(self.layer_1(inputs))
                x = self.batchnorm1(x)
                x = self.relu(self.layer_2(x))
                x = self.dropout(x)
                x = self.batchnorm2(x)
                x = self.relu(self.layer_3(x))
                x = self.batchnorm3(x)
                x = self.dropout2(x)
                x = self.relu(self.layer_4(x))
                x = self.batchnorm4(x)
                x = self.dropout2(x)
                x = self.layer_out(x)
return x
```

B.4. Other machine learning models

All machine learning classifiers other than the deep neural network classifier were built using scikit-learn. Protocols and code for each classifier can be found in the scikit-learn documentation at scikit-learn.org. [115]

All models were replicated from available GitHub repositories, referenced in each respective publication. [19]–[22]

B.5. Anisotropic network model

Anisotropic Network Models were built following the protocol created by the Computational and Systems Biology research group at the University of Pittsburgh. MATLAB [208] was used for all matrix manipulation and theoretical B-factor calculation based on all normal modes using the *eig* function. C code for ANM calculations was obtained from the ANM documentation from the Computational and Systems Biology research group at the University of Pittsburgh at http://anm.csb.pitt.edu/anmdocs/source.html. [184], [185], [244]

B.6. Protein structure modeling and analysis

All protein structure models, simulation including solvation and minimization, and intermolecular/intramolecular bond prediction were performed in UCSF ChimeraX. Protocol and code can be found through the UCSF Chimera documentation at https://www.cgl.ucsf.edu/chimerax/docs/user/index.html. [213]