Predicting cancer resistance mutations with protein-ligand graph networks

Robert Heeter

Department of Genomic Medicine, MD Anderson Cancer Center

Department of Bioengineering, Rice University

BIOE 401 Undergraduate Research

Summer Term 2022

Submitted 12 August 2022

Abstract

Drug resistance presents a significant obstacle to current targeted cancer therapies. Amino acid mutations that alter protein structure can adversely affect docking in the drug-target site, reducing treatment effectiveness. While numerous mutations have been documented, limited research has been done to analyze and predict the likelihood of these mutations occurring. On a separate note, the use of graph networks to model protein structures for machine learning and optimization applications is novel but promising. This research aims to use these graph networks in a pipeline designed to predict cancer mutations that may confer resistance. To train such a predictive model, simulated polypeptide mutations can be compared to their wild-type counterparts. The pipeline is still incomplete; the current program can model proteins as a graph network, and work is being done on mutation simulation. For this paper, human hemoglobin A2 and the epidermal growth factor receptor (EGFR) kinase domain with the gefitinib drug ligand are modeled as a proof-of-concept for the existing pipeline.

I. Introduction

Oncogenesis is the result of gene-level mutations that cause hyperactive gene expression. More than 1% of all genes in the human genome are known to be involved in possible cancer development[1]; notably, protein kinases, which are responsible for regulating a significant number of cellular pathways, are a common domain that is encoded by these cancer genes[1]. Epidermal growth factor receptor (EGFR) is an example of such a domain. Traditionally, "one-size-fits-all" cytotoxic treatments have been used to combat cancer development, affecting both cancer cells and normal body cells and thus producing strong side effects[2]. More recently, these generic non-selective approaches are being replaced by selective, or 'targeted', therapies that attack specific proteins and mechanisms to inhibit or mitigate cancer growth. For example, the drug gefitinib (also known as *Iressa*, produced by AstraZeneca) effectively targets and inhibits EGFR in some non-small-cell lung cancer patients, which led to its FDA approval along with similar drugs like erlotinib, and are shown to be more effective than traditional cytotoxic chemotherapy[2].

Targeted therapies still face numerous obstacles to their full effectiveness. For one, the selection of drug targets—proteins and other biomolecules that can dock a drug—remains a challenge, both in terms of researching specific druggable sites on cancer targets, and in terms of commercial and biological risk—investing in research on more documented and 'familiar' targets and ignoring more 'foreign' or unexplored targets (a manifestation of 'lemming behavior')[3]. Secondly, drug resistance can render a targeted therapy ineffective by preventing ligand docking and the necessary interactions with the target site. For the case of EGFR in non-small-cell lung cancer, gefitinib resistance can be acquired due to mutation in only a few years, leading to relapse[4]. For many targets, the range of all potentially resistance-conferring mutations is unclear[5]. While there is significant research into drug target discovery—to date,

over 1,500 FDA-approved drugs[6]—and work being done to minimize human bias in target selection through data-based pathways[7], there is limited research on a comprehensive data-driven approach to understanding drug resistance[5].

A better understanding of target mutations and their effect on drug therapies can be used to better treat patients who develop resistance and facilitate drug development. On the clinical side, mutational analysis can be used to more quickly identify secondary treatment options, while on the research side, the same analysis can be used to inform target discovery and therapy development. As a result, the use of combinatorial therapies—those which operate synergistically to further reduce side effects—may be easier and done with greater confidence[8], and the parallel development of drugs could be facilitated (**Figure 1**). For example, if a resistant mutant is determined to have a high likelihood of developing at a target, as concluded through mutational analysis, then variant therapies can be simultaneously designed to hinder resistance.

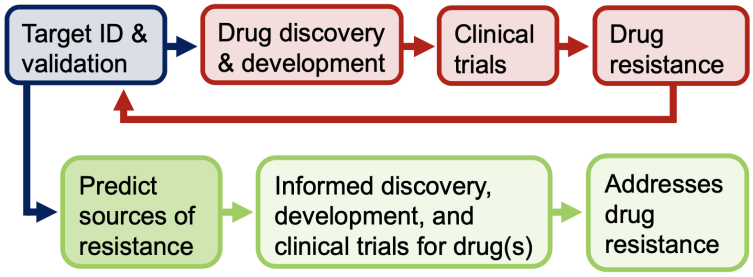

Figure 1. Simultaneous development of multiple drugs to handle resistance following resistance prediction (shown in green); red indicates traditional therapy development pathway.

This project aims to improve upon the work done by Martínez-Jiménez et al. (2017) in the design of a framework to identify single-point amino acid mutations that can lead to drug resistance[5]. While their model is well-validated and correctly detects critical protein mutations that align with experimental data (**Figure 2**), it is computationally complex as it relies on 58 and 89 features to be input into an amino acid-based and ligand-based random forest classifier,

respectively[5]. More specifically, this new pipeline implements a graph network representation of protein-ligand structures to describe their interactions. Here, atoms are represented as nodes in the network, and bonds and interactions are edges. Thus, mutations manifest as changes in the connectivity of the network, as both protein covalent bonds are added or removed, and binding interactions, such as hydrogen bonds and hydrophobic interactions are changed. Notably, attention-based graph neural networks gained traction when the company AlphaFold (part of Google and Alphabet) unveiled a system to predict 3D protein structures purely from sequence data with unprecedented accuracy[9]. However, AlphaFold's architecture is not designed to predict the effects of protein mutations, and while it is successful in determining a large-scale model for proteins, it has limited resolution at the atomic level when compared to experimental data in the Protein Data Bank[10]. These areas of improvement for both the framework by Martínez-Jiménez et al. and for AlphaFold's system are motivations for this project.

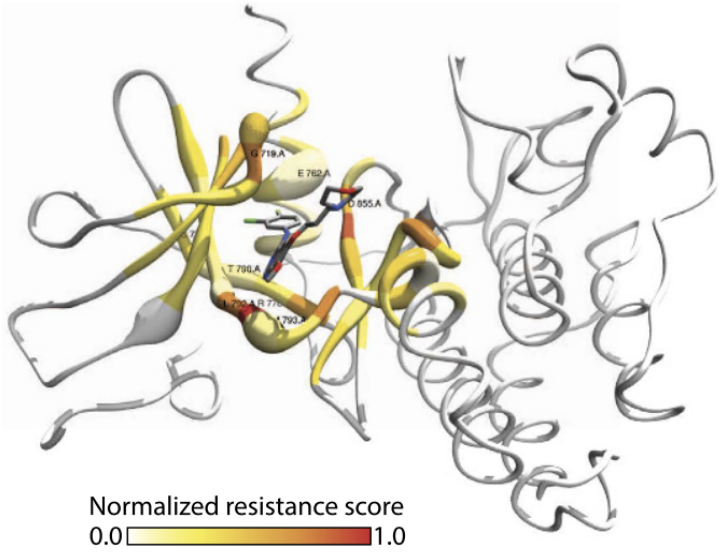

Figure 2. Visualization of the output ("resistance score") from the model produced by Martínez-Jiménez et al. for the EGFR kinase domain with gefitinib[5]; "the thickness of the ribbons indicates the accumulated mutational likelihood for that particular amino acid; the color represents the accumulated NRS [normalized resistance score, a rough metric for gauging the potential locations for resistance-conferring mutations]."

II. Dependencies

1. *Protein Data Bank*[11,12]

The Protein Data Bank (PDB) is a repository of experimental structural protein information and metadata summarized in *.pdb* files[11,12]. Each PDB file contains annotations, ligand information, secondary structure data, and atomic information, sorted by character position, and each line contains a separate record. The atomic information, used for this project, is shown in **Figure 3**. Note that some atoms may have multiple records in cases where their occupancy is uncertain (i.e. atoms #7-8 in **Figure 3**); in these cases, the first atom with an occupancy greater than or equal to 0.5 is used. The PDB also contains a separate HET Group Dictionary, which includes ligand information (i.e. connectivity) that is associated with a particular PDB file or protein.

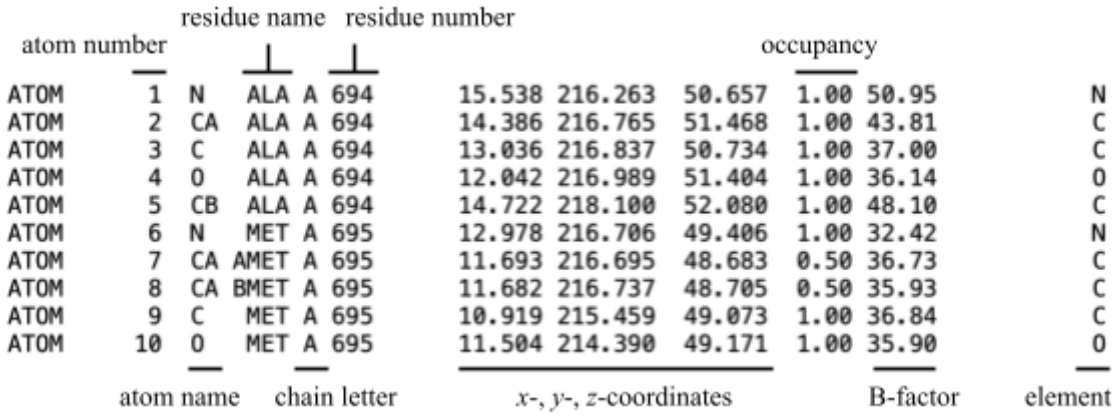

Figure 3. Protein Data Bank file atomic data structure.

2. *LigPlot*[13]

LigPlot is a program that calculates and visualizes protein-ligand interactions for an input PDB file using 3 sub-programs: *hbadd*, *hbplus*, and *ligplot*. *hbadd* is a program that determines ligand bond orders using the HET Group Dictionary discussed earlier, which are used in *hbplus*, an algorithm that identifies potential hydrogen bonds and hydrophobic non-bonded contacts using hydrogen donors and acceptors on the protein

polypeptide chain and ligand, relying on interatomic distances and bond angles[13]. As a default, hydrogen bonds are constrained to 2.70 to 3.35 Angstroms, and hydrophobic contacts are constrained to 2.90 to 3.90 Angstroms. This information is then used in *ligplot* to create a final schematic of the protein-ligand interface (**Figure 4**) and return a final list of calculated hydrogen bonds and hydrophobic interactions. In this project, LigPlot is run through Python using the OS for automation, rather than through its GUI.

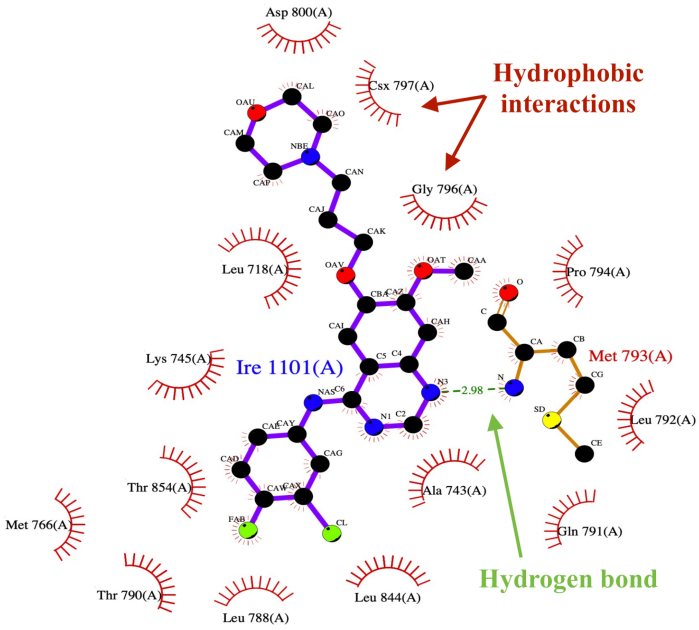

Figure 4. Example LigPlot output for the EGFR kinase domain with gefitinib. Hydrophobic interactions are shown as "eyelashes" and hydrogen bonds are shown as dashed lines. Note the singular hydrogen bond between IRE (gefitinib) atom *N3* and EGFR atom *N* on Methionine #793.

3. *MODELLER*[14]

MODELLER is a Python program used for homology and comparative modeling of protein structures in 3D[14]. This project passes a template sequence (corresponding to the PDB protein) aligned with a "mutant" target sequence into MODELLER, which optimizes protein energies and spatial restraints to iteratively return approximate models of the mutant protein. In this case, the "mutant" target sequence simply includes a single

amino acid substitution mutation (**Figure 5**). Ligands can either be included as an additional spatial restraint (optimization around the ligand) or removed.

```
>P1;4wkq_template
structure:4wkq_mod.pdb   :FIRST:@:  END: :::     :
AMGEAPNQALLRILKETEFKKIKVLGS/--GTVYKGLWIPEGEKVKIPVAIKE/----SPKANKEILDEAYVMAS
VDNPHVCRLLGICLTSTVQLITQLMPFG/LLDYVREHKDNIGSQYLLNWCVQIAKGMNYLEDRRLVHRDLAARNV
LVKTPQHVKITDFGLAKLLGAEEKEYHAEGGKVPIKWMALESILHRIYTHQSDVWSYGVTVWELMTFGSKPYDGI
PASEISSILEKGERLPQPPICTIDVYMIMVKCWMIDADSRPKFRELIIEFSKMARDPQRYLVIQGD/--------
-------------MDDVVDADEYLIPQ/.*

>P1;4wkq_target
sequence :4wkq_target:     : :     : :::     :
AMGEAPNQALLRILKETEFKKIKVLGS/--GTVYKGLWIPEGEKVKIPVAIKE/----SPKANKEILDEAYVMAS
VDNPHVCRLLGICLTSTVQLITQLWPFG/LLDYVREHKDNIGSQYLLNWCVQIAKGMNYLEDRRLVHRDLAARNV
LVKTPQHVKITDFGLAKLLGAEEKEYHAEGGKVPIKWMALESILHRIYTHQSDVWSYGVTVWELMTFGSKPYDGI
PASEISSILEKGERLPQPPICTIDVYMIMVKCWMIDADSRPKFRELIIEFSKMARDPQRYLVIQGD/--------
-------------MDDVVDADEYLIPQ/.*
```

Figure 5. MODELLER input alignment sequences. The first sequence is the template sequence, corresponding to the *'4wkq_mod.pdb'* file for the residues in the EGFR kinase domain with gefitinib. The second sequence is the target sequence with a methionine (M) #793 to tryptophan (W) substitution mutation.

4. *Other Dependencies*

While not yet complete, this project plans to use the OncoKB database of known resistance mutants to test the new pipeline. In addition, PyMol and Cytoscape are used briefly for visualization purposes.

III. Pipeline & Demo

1. *Protein-ligand graph network construction*

The first step in the pipeline is the construction of a protein-ligand graph network from a PDB input file (downloaded from the RCSB database), which lacks connectivity information for the protein, the ligand, and their interactions. While some tools, such as PyMol, can give connectivity information, they typically do so using rough atom proximity calculations, rather than via explicit chemical structures. Because the chemical structures for protein residues are known, they can be fit to the atomic data (i.e.

coordinates and elements); however, due to the large range of possible ligand structures, the same cannot easily be applied to the ligand bonds and protein-ligand interactions. The PDB input file is formatted and verified to ensure that the mapping of residue chemical structures to the atomic data is correct.

LigPlot is used to handle the ligand and binding bonds and interactions; it relies upon the same PDB input, the HET Group Dictionary (available from the RCSB database), and additional parameters for calculations. LigPlot returns multiple flies, three of which contain ligand covalent bond orders (*hbadd.bonds*), hydrogen bonds (*ligplot.hhb*), and hydrophobic interactions (*ligplot.nnb*) between the protein and ligand, which are returned to the network construction program and formatted accordingly.

Finally, the program creates the data structures to store the network information (see **Outputs & Discussion** section). First, the PDB file is parsed to produce an atom node list for every atom in the protein-ligand structure (*n* total atoms, so *n* rows in atom list). Each row contains atomic information, including atom element, atom residue, and atomic coordinates. Next, a square distance matrix (*n* rows by *n* columns) of pairwise distances between all atoms is computed using the Euclidean norm. Each row and column index of this matrix corresponds to an atomic row index in the atom list. A square connection matrix (adjacency matrix) of the same dimensions is also preallocated and filled with protein covalent backbone and residue bonds, following residue chemical structures and accounting for missing atom records in the PDB file. Additionally, ligand covalent bonds, hydrogen bonds, and hydrophobic interactions are added to the connection matrix using the LigPlot outputs. Finally, the square connection matrix

(adjacency matrix) is transformed into a bond tuple edge list, with additional information such as bond length and node data.

The atom list, distance matrix, connection matrix, and bond list are exported to *.csv* files to read. The atom list and bond list data are used in a Matplotlib color-coded 3D visualizer to verify the correct conversion of the PDB file to a graph network.

Figure 6 shows a generic example of the protein graph network construction process for the EGFR kinase domain with gefitinib (PDB code *4wkq*). The data and figures produced by the example code are shown in **Figures 7-9** and are explained more in the **Outputs & Discussion** section. Holistically, the graph network construction program is designed to be modular and allow for additional or variable inputs, such as user-defined binding interactions.

2. *Mutation simulation*

MODELLER is used for simulating numerous protein mutations to study the effect that particular mutations have on the overall protein structure. As previously described, using a target sequence, MODELLER makes amino acid additions, deletions, and substitutions to a template structure, which corresponds to a target sequence, and then optimizes spatial restraints (energies). So far, the pipeline includes functions to simulate specific mutations in MODELLER, which returns “mutant” PDB files that can be re-parsed. The MODELLER alignment sequences are very rigid, and the output mutant structures must also be validated by analyzing their energy profiles compared to their wild-type counterparts. Finally, more research must be done to determine whether the

ligand should be included in the binding site as a spatial restraint, or removed in the case that it might interfere with protein relaxation.

3. *Binding reassessment, network classifier, classifier validation*

Though they are still being developed, the next components of the prediction pipeline include the binding reassessment of the new mutant PDBs, classification of the mutant networks into effective and resistant proteins, and validation of the pipeline's predictions using test data. The binding effectiveness of the mutant PDBs can be determined in several possible ways, including LigPlot or other docking software, such as DeepChem or DockString, which both run through Python. In addition, docking software may be needed to optimize ligand orientation within the mutant binding site. Using multiple wildtype-mutant structures from MODELLER, a network classifier can (ideally) be trained to distinguish between effective and resistant models. Finally, as previously mentioned, the OncoKB database is one source of known resistance mutants that can be used to test the classifier for validation of the pipeline.

PROTEIN-LIGAND GRAPH NETWORK CONSTRUCTION

```
# PARAMETERS

# ligplot_program_path = path to LigPlot program
ligplot_program_path = "/Users/username/Path/LigPlus/lib/exe_mac64"

# ligplot_prm_file_path = path to LigPlot .prm file
ligplot_prm_file_path = "/Users/username/Path/LigPlus/lib/params/ligplot.prm"

# het_group_dictionary_file_path = path to HET Group Dictionary .cif file
het_group_dictionary_file_path = "/Users/username/Path/LigPlus/lib/params/components.cif"

# hbplus_params = 4 float parameters for hbplus thresholds
hbplus_params = [2.70, 3.35, 2.90, 3.90]

# pdb_url = PDB url to download .pdb file, with '[abcd]' in place of 4-character PDB code
pdb_url = "http://files.rcsb.org/download/[abcd].pdb"

# pdb_input = input [abcd] PDB code (if using url) or file path ending in [/abcd[...].pdb]
pdb_input = '4wkq' # EGFR kinase domain with gefitinib

# hetatm_name = target HETATM residue, with structure RESIDUE_NAME-CHAIN_ID, i.e. IRE-A
hetatm_name = ['IRE-A'] # gefitinib residue (on chain A)

# output_path = path to output folder
output_path = "/Users/username/Path/Output"


# PROGRAM

# Protein Data Bank
pdb_file_path,hetatm_name_list = get_pdb(pdb_input,pdb_url,output_path) # get PDB file
pdb_file_path = format_pdb(pdb_file_path,output_path) # format PDB file

# LigPlot
ligplot_hhb_file_path,ligplot_nnb_file_path,hbadd_bonds_file_path =
  get_ligplot(pdb_file_path,hetatm_name,ligplot_program_path,ligplot_prm_file_path,het_group_dictionary
  _file_path,hbplus_params,output_path) # run LigPlot

ligplot_hhb = format_ligplot(ligplot_hhb_file_path,hetatm_name) # format hydrogen bonds data
ligplot_nnb = format_ligplot(ligplot_nnb_file_path,hetatm_name) # format hydrophobic interactions data
hbadd_bonds = format_ligplot(hbadd_bonds_file_path,hetatm_name) # format ligand covalent bonds data

# Network Data Structures
atom_list = make_atom_list(pdb_file_path,hetatm_name_list) # organize atomic data

dist_matrix = make_distance_matrix(atom_list) # calculate bond lengths

conn_matrix = make_connection_matrix(atom_list)
conn_matrix = fill_protein_bonds(conn_matrix,atom_list) # add protein covalent bonds
conn_matrix = fill_ligand_bonds(conn_matrix,atom_list,ligplot_hhb) # add hydrogen bonds
conn_matrix = fill_ligand_bonds(conn_matrix,atom_list,ligplot_nnb) # add hydrophobic interactions
conn_matrix = fill_ligand_bonds(conn_matrix,atom_list,hbadd_bonds) # add ligand covalent bonds

bond_list = make_bond_list(conn_matrix,atom_list,dist_matrix) # make bond tuple edge list

# Export
export_atom_list(atom_list,output_path)
export_distance_matrix(dist_matrix,output_path)
export_connection_matrix(conn_matrix,output_path)
export_bond_list(bond_list,output_path)

# Plot
make_plot(bond_list,atom_list) # make 3D visualizer plot
```

Figure 6. Protein-ligand graph network construction process, written in JupyterLab.

IV. Outputs & Discussion

The graph network construction program produces four data structures containing node and edge information (**Figure 7**), and an optional 3D plot (**Figures 8-9**). The atom list includes each atom in the protein-ligand complex per row, after filtering through the PDB to remove unnecessary atoms (i.e. low-occupancy atoms or atoms in unspecified/additional ligands). The bond list contains the two node indices, bond type, bond length, and node information (not shown in the figure) on each row. The distance matrix is a square matrix of pairwise distances between atoms, organized by index; thus, there are zeros along the diagonal. The connection matrix is designed to be upper triangular and is quite sparse. Three-letter abbreviations are used to denote backbone (B), residue (R), or ligand (L) + covalent (C) + single (S), double (D), or triple (T) bonds; hydrogen bonds are denoted as 'HHH' and hydrophobic interactions are denoted as 'PPP'.

The two 3D plots in **Figure 8** are the outputs for a network of human hemoglobin with multiple protoporphyrins (left)[15] and the EGFR kinase domain with gefitinib (right)[16], showing how the program successfully handles multiple small molecules. The EGFR kinase domain with gefitinib also clearly shows the protein backbone (colored gray) starting at the top of the figure, with teal-colored residues extending outward. **Figure 9** offers a closer view of the gefitinib docking interactions and the bond coloring scheme. Notably, a singular hydrogen bond (orange) can be seen between gefitinib and methionine #793, which matches experimental information[15]. Methionine #793 lies in the "hinge" region of the kinase, between the two major lobes (left and right sides of the figure) where gefitinib binds[17]. **Figures 10-11** are 2D network representations of the bond list output from Cytoscape, colored using the same scheme as **Figure 9**. Again, the singular hydrogen bond is observable.

index	atom_numbe	atom_name	atom_type	residue_num	residue_nam	chain_id	x_coord	y_coord	z_coord	element
0	1	N	B	694	ALA	A	15.538	216.263	50.657	N
1	2	CA	B	694	ALA	A	14.386	216.765	51.468	C
2	3	C	B	694	ALA	A	13.036	216.837	50.734	C
3	4	O	B	694	ALA	A	12.042	216.989	51.404	O
4	5	CB	R	694	ALA	A	14.722	218.1	52.08	C

	atom_A	atom_B	bond_type	bond_length
2	0	1	BCS	1.49533508
3	1	2	BCS	1.54592949
4	1	4	RCS	1.5200898
5	2	3	BCD	1.2366673
6	2	8	BCS	1.33859628
7	4	5	RCS	1.50391955
8	5	6	RCS	1.25192971
9	5	7	RCD	1.24748307
10	8	9	BCS	1.47393826
11	9	10	BCS	1.53767552
12	9	12	RCS	1.5371041
13	10	11	BCD	1.22611296
14	10	22	BCS	1.330728
15	12	13	RCS	1.49808144

	A	B	C	D	E	F	G
1	0.00E+00	1.50E+00	2.53E+00	2.93E+00	2.46E+00	3.81E+00	4.40
2	1.50E+00	0.00E+00	1.55E+00	2.42E+00	1.52E+00	2.51E+00	3.10
3	2.53E+00	1.55E+00	0.00E+00	1.24E+00	2.51E+00	2.99E+00	3.75
4	2.93E+00	2.42E+00	1.24E+00	0.00E+00	3.12E+00	3.74E+00	4.70
5	2.46E+00	1.52E+00	2.51E+00	3.12E+00	0.00E+00	1.50E+00	2.37
6	3.81E+00	2.51E+00	2.99E+00	3.74E+00	1.50E+00	0.00E+00	1.25
7	4.40E+00	3.10E+00	3.75E+00	4.70E+00	2.37E+00	1.25E+00	0.00
8	4.60E+00	3.32E+00	3.28E+00	3.73E+00	2.36E+00	1.25E+00	2.21
9	3.62E+00	2.45E+00	1.34E+00	2.27E+00	3.35E+00	3.29E+00	3.73
10	4.85E+00	3.85E+00	2.48E+00	2.84E+00	4.72E+00	4.54E+00	4.99
11	6.03E+00	4.84E+00	3.52E+00	3.77E+00	5.33E+00	4.77E+00	5.19
12	7.16E+00	6.02E+00	4.66E+00	4.83E+00	6.56E+00	5.96E+00	6.31

	A	B	C	D	E	F	G
1		BCS					
2			BCS		RCS		
3				BCD			
4							
5						RCS	
6							RCS
7							
8							
9							
10							
11							
12							

Figure 7. Protein graph network data structures. Atom list (top) contains atomic information. Bond list (middle left) is a tuple edge list with bond information. Distance matrix (middle right) contains distances between all atoms. Connection matrix (bottom) is an adjacency matrix of bonds/interactions between specific atoms and aligns with the distance matrix.

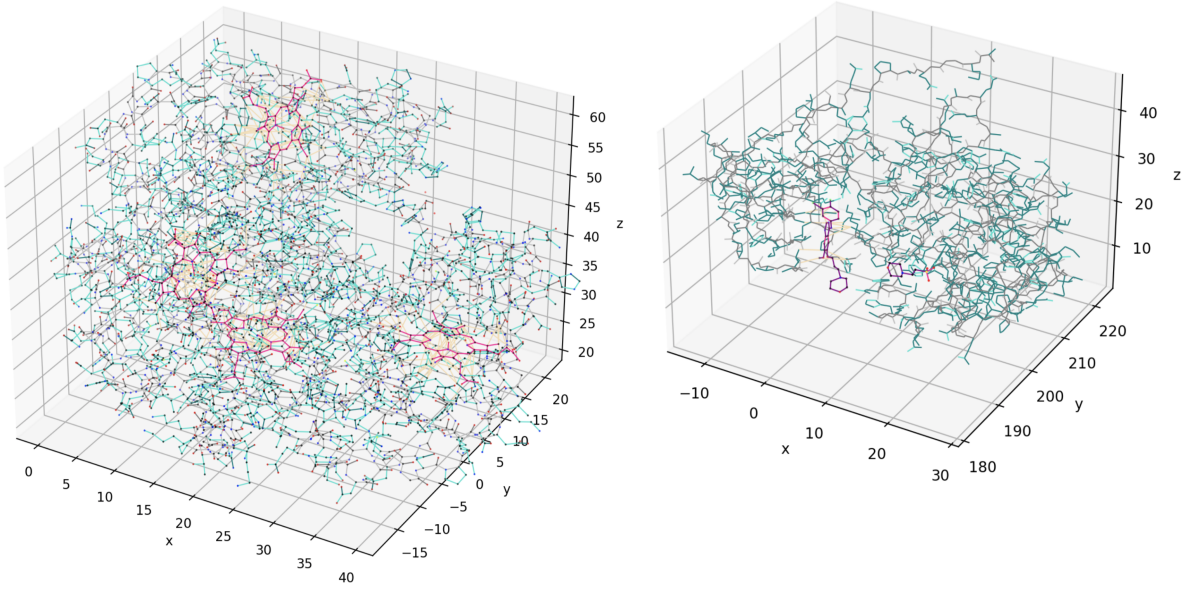

Figure 8. Human hemoglobin A2 structure (left) with multiple non-protein molecules (protoporphyrin with iron). EGFR kinase domain with gefitinib structure (right).

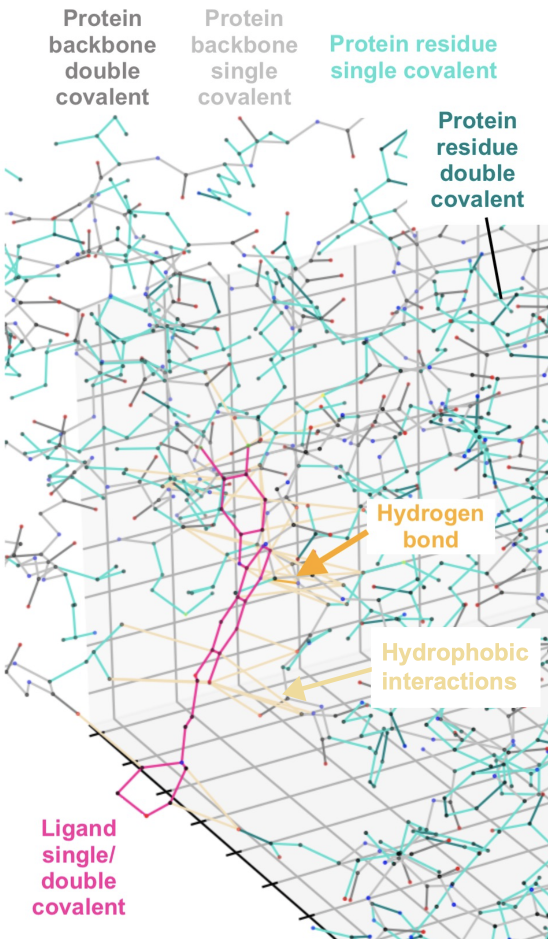

Figure 9. 3D plot of the EGFR kinase domain with gefitinib using Python's Matplotlib library from graph network data structures. Note the single hydrogen bond between a nitrogen on gefitinib and Met #793 on the EGFR kinase domain.

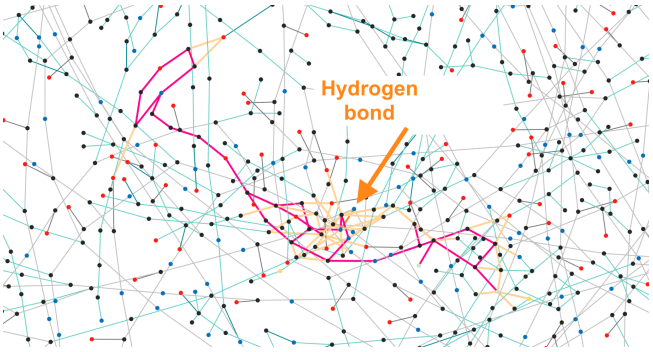

Figure 10. Local Cytoscape network model of the EGFR kinase domain with gefitinib.

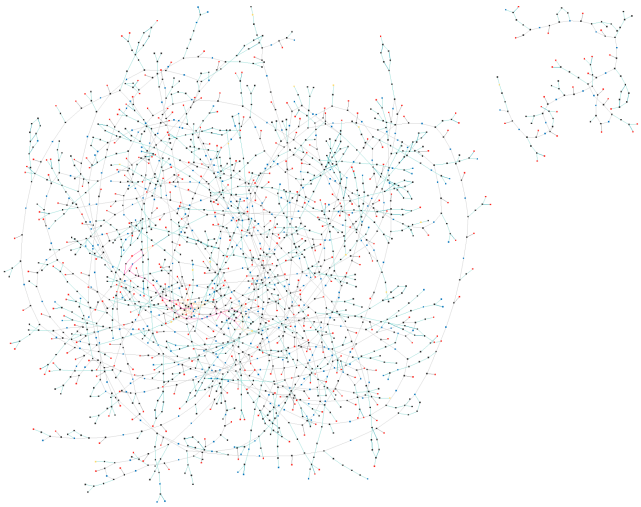

Figure 11. Full Cytoscape network model of the EGFR kinase domain with gefitinib. Note the two separate sections are due to a chain break from incomplete amino acid residue records.

V. Failure Analysis

One significant issue in the initial design of the protein graph network was the method used to assign bond orders to the ligand(s). PDB files do not contain ligand connectivity with bond order, so it must be determined through an alternate means. At first, RDKit and SMILES strings provided a potential solution; RDKit is a comprehensive cheminformatics package, which can easily be used to parse SMILES strings—strings of characters that encode the exact chemical

structure of a molecule. For example, "COC1=C(C=C2C(=C1)N=CN=C2NC3=CC(=C(C=C3)F)Cl)OCCCN4CCOCC4" is the SMILES string for gefitinib[16]. However, the SMILES strings stored in the PDB are not necessarily well-formed, and issues arose when RDKit detected chemistry "errors" and would not return coherent bonding information. Furthermore, the process of aligning the ligand atoms in the PDB to the bond information from RDKit was extremely complex, since the atoms are stored in an inconsistent order (i.e. "atoms 1-10" from the PDB do not directly correspond to "atoms 1-10" from RDKit). To circumvent this, the HET Group Dictionary was implemented instead, which is designed by and meant to correspond directly with the PDB.

More broadly, the PDB contains many edge cases and many considerations must be taken to properly interpret the data. Perhaps the largest obstacle in developing the network was the issues arising with mapping data (such as LigPlot bonds) to the PDB.

VI. Conclusions & Next Steps

Altogether, the current pipeline can produce a protein graph network given an input PDB file, using LigPlot to determine ligand covalent bonds, hydrogen bonds, and hydrophobic interactions, and simulate simple substitution mutations on the PDB file using MODELLER. So far, the outputs appear to be valid and robust, though more testing should be done with a larger set of protein-ligand complexes. Looking forward, the wild-type and mutant proteins need to be reassessed to train a classifier that can distinguish between effective and resistant networks.

VII. References

1. Futreal PA, Coin L, Marshall M, et al. A census of human cancer genes. *Nat Rev Cancer*. 2004;4(3):177-183. doi:10.1038/nrc1299
2. Mayekar MK, Bivona TG. Current landscape of targeted therapy in lung cancer. *Clin Pharmacol Ther*. 2017;102(5):757-764. doi:10.1002/cpt.810
3. Booth B. Cancer drug targets: the march of the lemmings. https://www.forbes.com/sites/brucebooth/2012/06/07/cancer-drug-targets-the-march-of-the-lemmings/. Accessed August 9, 2022.
4. Kobayashi S, Boggon TJ, Dayaram T, et al. *EGFR* mutation and resistance of non–small-cell lung cancer to gefitinib. *N Engl J Med*. 2005;352(8):786-792. doi:10.1056/NEJMoa044238
5. Martínez-Jiménez F, Overington JP, Al-Lazikani B, Marti-Renom MA. Rational design of non-resistant targeted cancer therapies. *Sci Rep*. 2017;7(1):46632. doi:10.1038/srep46632
6. Santos R, Ursu O, Gaulton A, et al. A comprehensive map of molecular drug targets. *Nat Rev Drug Discov*. 2017;16(1):19-34. doi:10.1038/nrd.2016.230
7. Al-Lazikani B, Workman P. Minimizing bias in target selection by exploiting multidisciplinary big data and the protein interactome. *Future Medicinal Chemistry*. 2016;8(14):1711-1716. doi:10.4155/fmc-2016-0133
8. Al-Lazikani B, Banerji U, Workman P. Combinatorial drug therapy for cancer in the post-genomic era. *Nat Biotechnol*. 2012;30(7):679-692. doi:10.1038/nbt.2284
9. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*. 2021;596(7873):583-589. doi:10.1038/s41586-021-03819-2
10. Callaway E. What's next for AlphaFold and the AI protein-folding revolution. *Nature*. 2022;604(7905):234-238. doi:10.1038/d41586-022-00997-5
11. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. *Nucleic Acids Research*. 2000;28(1):235-242. doi:10.1093/nar/28.1.235
12. Berman H, Henrick K, Nakamura H. Announcing the worldwide Protein Data Bank. *Nat Struct Mol Biol*. 2003;10(12):980-980. doi:10.1038/nsb1203-980
13. Laskowski RA, Swindells MB. Ligplot+: multiple ligand–protein interaction diagrams for drug discovery. *J Chem Inf Model*. 2011;51(10):2778-2786. doi:10.1021/ci200227u
14. Šali A, Blundell TL. Comparative protein modeling by satisfaction of spatial restraints. *Journal of Molecular Biology*. 1993;234(3):779-815. doi:10.1006/jmbi.1993.1626
15. Sen U, Dasgupta J, Choudhury D, et al. Crystal structure of Human hemoglobin A2 (in R2 state) at 2.2 A resolution. *Worldwide Protein Data Bank*. 2004. doi:10.2210/pdb1si4/pdb
16. Yosaatmadja Y, & Squire CJ. 1.85 angstrom structure of EGFR kinase domain with gefitinib. *Worldwide Protein Data Bank*. 2014. doi:10.2210/pdb4wkq/pdb

17. Yun CH, Boggon TJ, Li Y, et al. Structures of lung cancer-derived egfr mutants and inhibitor complexes: mechanism of activation and insights into differential inhibitor sensitivity. *Cancer Cell*. 2007;11(3):217-227. doi:10.1016/j.ccr.2006.12.017

VIII. Acknowledgements

I would like to thank Dr. Stephanie Schmidt for her support and guidance during my summer experience at MD Anderson in the A3D3a program. In particular, her work to outline my project's goals, and flexibility and availability to hold meetings helped me grasp and tackle some of the more complex aspects of my program. In addition, I am immensely grateful to Professor Bissan Al-Lazikani; her wisdom and expertise in computer science and structural biology was instrumental in pointing me in the right direction and recommending helpful resources and papers. I would also like to thank Dr. Phillip Gingrich for his thoughtful input, inquiry, and feedback on my project towards the end of my summer experience. Finally, I would like to thank my peers—and now friends—Carl Ho and Luca Cohen for their generous advice and insight on the smaller details of my project.