
Jason Eden Sanchez
The University of Texas at El Paso

Follow this and additional works at: https://scholarworks.utep.edu/open_etd

Part of the Artificial Intelligence and Robotics Commons, and the Pharmacy and Pharmaceutical Sciences Commons

Recommended Citation

This is brought to you for free and open access by ScholarWorks@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of ScholarWorks@UTEP. For more information, please contact lweber@utep.edu.
COVID SYNERGY: A MACHINE LEARNING APPROACH UNCOVERING
POTENTIAL TREATMENT COMBINATIONS FOR SARS-COV-2

JASON EDÉN SÁNCHÉZ OROZCO

Master’s Program in Computational Science

APPROVED:

Suman Sirimulla, Ph.D., Chair

Natasha Sharma, Ph.D.

Jeri Sias, PharmD.

Stephen Crites, Ph.D.
Dean of the Graduate School
Para mi familia,

con cariño
COVID SYNERGY: A MACHINE LEARNING APPROACH UNCOVERING
POTENTIAL TREATMENT COMBINATIONS FOR SARS-COV-2

by

JASON EDÉN SÁNCHEZ OROZCO, BS

THESIS
Presented to the Faculty of the Graduate School of
The University of Texas at El Paso
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

Master's Program in Computational Science
THE UNIVERSITY OF TEXAS AT EL PASO
May 2022
Acknowledgements

Science is a team effort. It would be incredibly disingenuous of me to believe I could do my work alone. Research is truly not done by individuals; rather, it is supported by communities. That is why I want to take time to acknowledge those who have helped me in my academic career and personal life.

**Papá y Mamá**: You have always encouraged me to pursue higher education. Without your love and support, I don’t know if I could have gone as far as I have. I remember those late nights when I would be coding and you guys would pop into my room and bring me a snack or just check in on me. Those little gestures go a long way for us grad students.

**Noel, Gema, and Jonathan**: You guys are the best siblings anyone could ask for. Many people have siblings, but not many can call their siblings best friends. You have been there for me through thick and thin and I love all of you dearly.

**Bella**: You put a smile on my face pretty much every time we get together. Your friendship means more to me than all of the money in the world. I know these past two years have put a wrench in a lot of our plans but I cherish all those memories we made—even when those memories were just us social distancing (playing Fortnite) until the wee hours of the morning.

**Dorian**: You are one of the most genuine, supportive, and true friends anyone could ask for. There is never a dull moment with you—remember Ruidoso? I can’t wait to see what life has in store for us; regardless, I know I can count on you for anything.

**Dr. Sirimulla**: Thank you for being my advisor and thank you for always pushing me to realize my potential. Working with you, I can honestly say you helped revitalize my love for science.

**My entire lab**: All of you have helped me so much—either directly in my research or simply by brightening my day. Your efforts don’t go unnoticed. Wherever life takes you, I hope you find success and happiness.
Abstract

For more than two years, the COVID-19 pandemic has upended the lives of billions of individuals worldwide leading to disruptions in healthcare, the economy and society at large. As the pandemic enters its third year, the human impact cannot be overstated and the need to develop effective pharmaceuticals remains. Though there currently exits FDA-approved medications for COVID-19, the emergence of novel variants, such as Omicron, highlights the importance of discovering new therapies which will continue to be effective regardless of the pandemic’s progression. Because discovering new medications is a costly and time-intensive endeavor, my approach entails drug repurposing to test medications which are already in use. In this publication, combinations of previously approved drugs are tested for synergy against SARS-CoV-2. The intention of using combinations of drugs is to improve patient outcomes and prevent treatment escape. My approach uses various machine learning models to predict synergy for repurposed drugs which have been previously shown to have activity against SARS-CoV-2. Drug synergy models are made publicly available to researchers hoping to study SARS-CoV-2. In addition to the in silico experiments, top-scoring combinations are experimentally validated in vitro.
# Table of Contents

<table>
<thead>
<tr>
<th>Acknowledgements</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>x</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
</tbody>
</table>

## Chapter

1. Introduction                                       | 1   |

  1.1 SARS-CoV-2 Virology                              | 1   |
    1.1.1 Structure                                     | 2   |
    1.1.2 Genome and Life Cycle                         | 3   |
    1.1.3 Pathophysiology                               | 8   |
    1.1.4 Variants                                      | 12  |

  1.2 SARS-CoV-2 Pharmaceuticals                        | 14  |
    1.2.1 Vaccines                                     | 15  |
    1.2.2 Antivirals                                    | 17  |
    1.2.3 Immune System Modulators                      | 19  |

2. Background                                         | 21  |

  2.1 Molecular Descriptors                            | 21  |
    2.1.1 Molecular Descriptors Used                   | 23  |
      2.1.1.1 RDKit Descriptors                         | 23  |
      2.1.1.2 Extended Connectivity Fingerprints        | 24  |

  2.2 Machine Learning                                  | 25  |
    2.2.1 Algorithms Used                               | 30  |
      2.2.1.1 Random Forest (RF)                         | 30  |
      2.2.1.2 Multi-layer Perceptron (MLP)               | 32  |
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.2</td>
<td>Hyperparameter Optimization</td>
<td>34</td>
</tr>
<tr>
<td>2.2.2.1</td>
<td>RF Hyperparameters</td>
<td>34</td>
</tr>
<tr>
<td>2.2.2.2</td>
<td>MLP Hyperparameters</td>
<td>35</td>
</tr>
<tr>
<td>2.2.2.3</td>
<td>Grid Search</td>
<td>35</td>
</tr>
<tr>
<td>2.2.2.4</td>
<td>K-Fold Cross Validation</td>
<td>36</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Evaluation Metrics</td>
<td>37</td>
</tr>
<tr>
<td>2.3</td>
<td>Synergy</td>
<td>40</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Bliss Independence</td>
<td>41</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Cokol β</td>
<td>43</td>
</tr>
<tr>
<td>3</td>
<td>Methodology</td>
<td>44</td>
</tr>
<tr>
<td>3.1</td>
<td>Data Preparation</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Molecular Descriptors</td>
<td>44</td>
</tr>
<tr>
<td>3.3</td>
<td>Machine Learning</td>
<td>45</td>
</tr>
<tr>
<td>3.4</td>
<td>Combination Assays</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>Results and Discussion</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>Machine Learning Models</td>
<td>49</td>
</tr>
<tr>
<td>4.2</td>
<td>Biological Validation</td>
<td>51</td>
</tr>
<tr>
<td>4.3</td>
<td>Mechanisms of Action</td>
<td>56</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Mebydrolin</td>
<td>56</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Fingolimod</td>
<td>57</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Melitracen</td>
<td>58</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion and Future Work</td>
<td>59</td>
</tr>
<tr>
<td>5.1</td>
<td>Conclusion</td>
<td>59</td>
</tr>
<tr>
<td>5.2</td>
<td>Specific Aims</td>
<td>59</td>
</tr>
<tr>
<td>5.3</td>
<td>Potential Solution</td>
<td>60</td>
</tr>
<tr>
<td>5.3.0.1</td>
<td>Compiling Mechanistic Information</td>
<td>60</td>
</tr>
<tr>
<td>5.3.0.2</td>
<td>Generating 3D descriptors</td>
<td>60</td>
</tr>
<tr>
<td>5.3.0.3</td>
<td>Using Graph-Based Machine Learning on PyG</td>
<td>61</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.4</td>
<td>Time Schedule</td>
<td>61</td>
</tr>
<tr>
<td>6</td>
<td>Supplemental Information</td>
<td>63</td>
</tr>
<tr>
<td>6.1</td>
<td>RDkit Descriptors List</td>
<td>63</td>
</tr>
<tr>
<td>6.2</td>
<td>RDkit Descriptor Families</td>
<td>64</td>
</tr>
<tr>
<td>6.3</td>
<td>Non-Synergistic Drug Combinations</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>Curriculum Vita</td>
<td>110</td>
</tr>
</tbody>
</table>
# List of Tables

1.1 Summary of SARS-CoV-2 Proteins. ................................................. 7

1.2 COVID-19 Disease Classification. .................................................. 11

1.3 COVID-19 Variants of Concern. ...................................................... 14

1.4 Summary of COVID-19 Vaccines. ..................................................... 17

1.5 Summary of COVID-19 Antivirals. ................................................... 18

1.6 Summary of COVID-19 Immune System Modulators. ........................... 20

2.1 Important Features from Animal Data. ............................................... 27

2.2 Animal Data in Augmented Matrix Form. .......................................... 28

2.3 Animal Data Train Set. ................................................................. 29

2.4 Animal Data Test Set. ................................................................. 29

2.5 Animal Data True Labels and Predictions. ........................................ 30

2.6 Random Forest Hyperparameter Grid. .............................................. 36

2.7 Multi-layer Perceptron Hyperparameter Grid. .................................... 36

2.8 Example Confusion Matrix. .......................................................... 39

2.9 Example Dose Response Matrix. .................................................... 41

2.10 Example ΔBliss Matrix. ............................................................... 43

3.1 Classification Thresholds. .............................................................. 46

4.1 Top Scoring Combinations. ............................................................ 50

4.2 Experimental β. ............................................................................ 55

5.1 Tentative Timeline for Future Work. .................................................. 62
List of Figures

1.1 SARS-CoV-2 Structure and Genome. [21] ........................................... 3
1.2 Synthesis of Subgenomic mRNA. [26] ............................................... 5
1.3 SARS-CoV-2 Life Cycle. [27] ................................................................. 6
1.4 COVID-19 Pathophysiology. [68] ......................................................... 10
1.5 SARS-CoV-2 Pharmaceuticals. ............................................................. 16

2.1 Cartoon Representation of a Molecular Descriptor. [132] ....................... 23
2.2 Labeled Animal Data. ............................................................................. 26
2.3 Example Random Forest. ....................................................................... 32
2.4 Example Multi-layer Perceptron ............................................................ 34
2.5 3-Fold Cross Validation Example. ......................................................... 37

3.1 Training Feature Vector Pairs. ............................................................... 45
3.2 Testing Feature Vector Pairs. ................................................................. 47

4.1 Test Set Classification Metrics. .............................................................. 51
4.2 External Dataset Classification Metrics. ............................................... 52
4.3 Combination 4 Percent Cell Survival. .................................................. 53
4.4 Combination 4 ΔBliss. ........................................................................... 53
4.5 Combination 7 Percent Cell Survival. .................................................. 54
4.6 Combination 7 ΔBliss. ........................................................................... 54
4.7 Mebhydrolin Chemical Structure. ......................................................... 57
4.8 Fingolimod Chemical Structure. ......................................................... 57
4.9 Melitracen Chemical Structure. ............................................................ 58

xi
Chapter 1

Introduction

1.1 SARS-CoV-2 Virology

On March 11, 2020 the World Health Organization (WHO) declared COVID-19 a pandemic. SARS-CoV-2, the virus which causes Coronavirus Disease 2019 (COVID-19), would subsequently cause hundreds of millions of infections globally. At the time of this publication, more than five million individuals around the world had lost their lives to COVID-19. [1] Indeed, the economic and societal upheaval caused by the virus is still being felt in 2022. Although there exists vaccine and treatment options for COVID-19, the threat of new variants warrants the continued discovery of novel agents to combat SARS-CoV-2.

Understanding the virology of SARS-CoV-2 is a critical first step in developing treatments for the virus. SARS-CoV-2 is colloquially referred to as a coronavirus. [2] Formally, coronaviruses are members of the family *Coronaviridae* and subfamily *Orthocoronavirinae*. [3] Other notable coronaviruses include SARS-CoV and MERS-CoV—the viruses responsible for Severe Acute Respiratory Syndrome (SARS) and Middle Eastern Respiratory Syndrome (MERS), respectively. These three viruses are known to infect bronchial epithelial cells, pneumocytes, and upper respiratory tract cells, leading to the aforementioned pathologies. [3] Herein, a basic overview of the virology of coronaviruses will provide a starting point for a discussion of potential drug targets. Although much of this information applies to all coronaviruses, special attention will be given to aspects of SARS-CoV-2 which may be exploited for treatment.
1.1.1 Structure

Coronaviruses are enveloped viruses with RNA packaged about Nucleocapsid (N) proteins (see Figure 1.1). Embedded within the envelope are the Spike (S), Membrane (M), and Envelope (E) proteins. Some coronaviruses—not including SARS-CoV-2—have additional Hemagglutinin-Esterase (HE) dimer proteins in the viral envelope. The envelope is derived from the host cell membrane. As we will see in Section 4.3, medications which modify the host cell membrane might be able to hinder SARS-CoV-2 activity. Inside the envelope, RNA and N proteins form ribonucleoprotein (RNP) structures which serve to package the genome in what is termed the viral capsid. [4, 5, 6]

The spike protein (S) is the principal component facilitating entry into a host cell. Studies have shown that S binds to Angiotensin-Converting Enzyme 2 (ACE2) in human host cells. [7, 8, 9, 10] Naturally, compounds which disrupt S binding to ACE2—by interacting with S, ACE2 or both—are potential antivirals for SARS-CoV2. Importantly, upon S binding, the virus is cleaved by host proteases to facilitate membrane fusion. [11, 12] Two such proteases, Furin and TMPRSS2, are responsible for different stages of the viral fusion process. [13, 10, 14] Given their role in promoting viral entry, these human proteins are also prospective drug targets.

The membrane (M) protein along with the envelope (E) protein work together during SARS-CoV-2 replication to promote viral assembly. [15] The work done by Boson et al. demonstrates E and M regulate cellular localization and post-transnational modification of S. Indeed, expression of E and M were shown promote intracellular retention of S, keeping the spike protein close to the viron assembly site. Additionally, E and M also influence the N-glycosylation of S—a key modification required for S to bind to ACE2. [16] Taken together, these observations suggest E and M may also be targeted for COVID-19 therapies.

N proteins are responsible for housing SARS-CoV-2 RNA. N proteins are able to protect RNA, enable mature viron formation, and dissociate from RNA when it is needed for translation and replication. [4] Depending on the progression of the viral life cycle, N proteins must balance the need to bind tightly and dissociate to RNA. Ligands which disrupt this
RNA binding could be investigated as potential therapeutics. Another important remark is that N has been shown to interact specifically with M through the carboxyl tail of M. [17, 18, 19, 20] Disrupting this interaction might also prove to be a way for ligands to treat COVID-19.

![SARS-CoV-2 Structure and Genome](image)

Figure 1.1: SARS-CoV-2 Structure and Genome. [21]

### 1.1.2 Genome and Life Cycle

SARS-CoV-2 has a positive-sense, single-stranded, RNA genome (+gRNA) of about 30,000 bases. [22] The genome codes for 16 non-structural proteins (nsps), 4 structural proteins (described in Section 1.1.1), and 9 accessory proteins. [23] In brief, nsps aid in viral
transcription and replication, structural proteins form the body of the virus, and accessory proteins assist in evading the host immune response or contributing to the pathogenicity of SARS-CoV-2.

Viral infection begins when S attaches to ACE2 receptors leading to fusion at the host membrane or viral uptake by endosomes (step 1 in Figure 1.3). Following entry, the viral genome is released into the cytosol (step 2) and directly translated by host cell ribosomes (step 3) which is a hallmark of positive-sense viruses. Two polypeptides are produced from this initial translation: pp1a and pp1ab. These two products, differentially produced by –1 ribosomal frameshifting, contain all viral nsps. [24]

Two nsps have proteolytic activity which serve to cleave pp1a and pp1ab into independent nsps. The first of these, nsp3, separates nsp1/nsp2, nsp2/nsp3 and nsp3/nsp4 (see the brown arrows in Figure 1.1). The second, nsp 5, makes all the remaining cleavages. Following polyprotein processing, the individual nsps perform their RNA replication and transcription functions on +gRNA (step 4). Together, the nsps are known as the viral replication and transcription complex (RTC). Here, I highlight key processes important to coronavirus RNA replication.

Nsps can be grouped into 3 different categories according to their function. Nsp1 degrades host mRNA, thereby increasing the translation of viral proteins. Nsp 12-16 together form the machinery which performs RNA synthesis, RNA proofreading and RNA modification with nsp 12 acting as the RNA-dependent RNA Polymerase (RdRP). Nsp 2-11 perform auxiliary functions which assist the core transcription and replication activities of nsp 12-16. Table 1.1 has information about SARS-CoV-2 proteins and their specific functions. Here a detailed discussion of each individual protein is not done. Instead the interested reader is encouraged to look at the table as a starting point for the many functions required for SARS-CoV-2 viral replication. For our purposes, it suffices to say nsps work together to make negative-sense RNA from the viral genome.

These negative-sense copies are transcribed starting from the polyadenylation site (polyA) of the viral genome (see Figure 1.1). A characteristic feature of coronavirus RNA replication
is that the +gRNA template strand can give rise to subgenomic or full-length negative-sense RNA (-sgRNA or -gRNA, respectively). [25] Full-length negative-sense RNA production occurs exactly as expected: the RTC reads through the entire +gRNA starting at the polyA site to produce complimentary -gRNA. Of course, the resulting -gRNA is reverse transcribed to make a copy of +gRNA.

The production of subgenomic RNA occurs through a process called discontinuous viral transcription. [25] As the RTC produces nascent negative-sense RNA, it encounters transcription regulatory sequences (TRSs) on the +gRNA. These sequences can prompt transcription to skip ahead to the leader sequence on the template strand (see Figure 1.2). This produces a set of different length -sgRNAs which all begin with the same 5′ sequences. When -sgRNAs are reverse transcribed, they form a set of nested 3′ co-terminal positive-sense subgenomic RNAs (+sgRNAs). Although the exact number of canonical sgRNAs is debated in the scientific community, we show ORFs 2-10 in Table 1.1 for completeness. Figure 1.1 depicts 9 +sgRNAs and highlights how these transcripts are nested and share a common 3′ end.

Figure 1.2: Synthesis of Subgenomic mRNA. [26]

From +sgRNAs, host ribosomes translate these into structural (S,E, M, or N proteins) or accessory proteins (step 5). We have already discussed the function of structural proteins
and the reader is encouraged to view Table 1.1 to get a sense of what accessory proteins do. The crucial idea is that N associates with +gRNA in the cytoplasm to form the nucleocapsid. Meanwhile, the other structural proteins are translated embedded in the Endoplasmic Reticulum (ER) where they will form part of the viral membrane. As the ER buds, the structural proteins form interactions with the nucleocapsid which is internalized inside the nascent viral particle (step 6). Viral assembly continues in outgoing vesicles which eventually reach the host cell membrane (step 7). The release of viral particles into the extracellular environment marks the end of the viral life cycle (step 8).

Figure 1.3: SARS-CoV-2 Life Cycle. [27]
Table 1.1: Summary of SARS-CoV-2 Proteins.

<table>
<thead>
<tr>
<th>ORFs</th>
<th>Protein</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORF 1a ORF 1b</td>
<td>nsp1</td>
<td>Promotes host mRNA degradation [28, 29]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Counters innate immune response [30]</td>
</tr>
<tr>
<td></td>
<td>nsp2</td>
<td>Function unknown</td>
</tr>
<tr>
<td></td>
<td>nsp3</td>
<td>Mediates pp1a and pp1ab proteolytic cleavage [31]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Has de-ADP-ribosylation activity [32]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Deubiquitination activity [33, 34]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Anti-interferon activity [35]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Promotes DMV formation [36, 37, 38, 39]</td>
</tr>
<tr>
<td></td>
<td>nsp4</td>
<td>Interacts with cellular membranes [36, 37, 38, 39]</td>
</tr>
<tr>
<td>nsp5 (M\text{PRO})</td>
<td></td>
<td>Main protease for pp1a and pp1ab cleavage [31]</td>
</tr>
<tr>
<td></td>
<td>nsp6</td>
<td>Promotes DMV formation [36, 37, 38, 39]</td>
</tr>
<tr>
<td></td>
<td>nsp7</td>
<td>Cofactor for RNA polymerase complex [24, 40, 41]</td>
</tr>
<tr>
<td></td>
<td>nsp8</td>
<td>Cofactor for RNA polymerase complex [24, 40, 41, 42]</td>
</tr>
<tr>
<td></td>
<td>nsp9</td>
<td>Nucleic-acid binding protein [43]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implicated to influence virulence [44]</td>
</tr>
<tr>
<td></td>
<td>nsp10</td>
<td>Cofactor for nsp14 and nsp16 [45]</td>
</tr>
<tr>
<td>ORF 1a</td>
<td>nsp11</td>
<td>Function unknown</td>
</tr>
<tr>
<td>ORF 1b</td>
<td>nsp12</td>
<td>RNA-dependent RNA polymerase [24]</td>
</tr>
<tr>
<td></td>
<td>nsp13</td>
<td>Helicase and ATPase activity [46]</td>
</tr>
<tr>
<td></td>
<td>nsp14</td>
<td>Exonuclease activity [47]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RNA cap formation [48]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Guanosine N7-methyltransferase activity [48]</td>
</tr>
<tr>
<td></td>
<td>nsp15</td>
<td>Endoribonuclease activity [49]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Helps virus evade innate immunity [50]</td>
</tr>
<tr>
<td></td>
<td>nsp16</td>
<td>RNA cap formation [51]</td>
</tr>
<tr>
<td>ORF</td>
<td>Description</td>
<td>Function</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ORF 2</td>
<td>2'-O-methyltransferase activity [51]</td>
<td>S Binds to ACE2 [7, 8, 9, 10] Initiates viral entry into host cells [7, 8, 9, 10]</td>
</tr>
<tr>
<td>ORF 3a</td>
<td>Orf3a Induces apoptosis and viral release [23]</td>
<td></td>
</tr>
<tr>
<td>ORF 3b</td>
<td>Orf3b Influences activation of AP-1 [23]</td>
<td></td>
</tr>
<tr>
<td>ORF 6</td>
<td>Orf6 Inhibits host cell translation [52]</td>
<td></td>
</tr>
<tr>
<td>ORF 7a</td>
<td>Orf7a Interacts with structural proteins and Orf3a [52]</td>
<td></td>
</tr>
<tr>
<td>ORF 7b</td>
<td>Orf7b Function unknown</td>
<td></td>
</tr>
<tr>
<td>ORF 8</td>
<td>Orf8 Suspected to help virus evade immune response [53]</td>
<td></td>
</tr>
<tr>
<td>ORF 9a</td>
<td>N Binds virus RNA to form the nucleocapsid [4]</td>
<td></td>
</tr>
<tr>
<td>ORF 9b</td>
<td>Orf9b Helps virus evade immune response [54]</td>
<td></td>
</tr>
<tr>
<td>ORF 10</td>
<td>Orf10 Function unknown</td>
<td></td>
</tr>
</tbody>
</table>

1.1.3 Pathophysiology

ACE2 receptors are found in a wide range of organ systems in the human body although they are heavily expressed in lung alveolar epithelial cells—entry into which initiates the respiratory pathologies of COVID-19. [55] Still, there exists many other cell types which SARS-CoV-2 may infect. In fact, SARS-CoV-2 has been known interact with the cardiovascular system, gastrointestinal system, excretory system, integumentary system, and even nervous system. [56, 57] Although the wide-ranging effects of SARS-COV-2 is an active area of research, I will focus on viral pathophysiology as it relates to the respiratory
system and immune system because these are linked to the majority of severe illness and hospitalization events.

COVID-19 may present with a variety of respiratory symptoms, some of which include: fever, dry cough, shortness of breath (dyspnea), rhinitis, and fatigue. [56, 58, 59, 60]. Although the majority of cases are mild, about 14% of cases lead to severe pneumonia and 5% lead to acute respiratory distress syndrome (ARDS), sepsis, or multisystem organ failure (MOF) for the original variant. [22]

COVID-19 pathogenesis begins when SARS-CoV-2 enters into susceptible cells through ACE2 receptors in the upper and lower respiratory system, typically via large respiratory droplets. [55] Following infection, viral replication inside host cells continues, eventually leading to cell death. [61, 62] Markers of infection such as damage-associated molecular patterns (DAMPs) and pathogen-associated molecular patterns (PAMPs) are subsequently released and recognized by Toll-like receptors (TLRs) and NOD-like receptors (NLRs) from neighboring epithelial cells and macrophages. [63, 64] These events lead to upregulation of the production of pro-inflammatory cytokines and chemokines and the subsequent recruitment of more immune cells such as macrophages, monocytes, and T cells (see Figure 1.4). For the majority of individuals infected with SARS-CoV-2, the response of these cells is appropriate and leads to viral clearance and patient recovery. However, for some individuals it has been postulated that this positive feedback loop continues and leads to an overactive immune response—often dubbed the "cytokine storm." [65, 66, 67] This cytokine storm results in the pulmonary damage associated with severe or critical COVID-19 (See Table 1.2).
In developing effective treatments for COVID-19 it may be worth characterizing potential therapies into two broad categories: anti-inflammatory drugs and anti-virals. Thus, a sound strategy for stopping the progression of COVID-19 requires halting the replication of SARS-CoV-2 early during the course of infection and/or treating the overactive immune response if it occurs. An overview of available therapies is discussed later in this chapter.

It is worth noting that the vast majority of COVID-19 cases do not result in life-threatening pathologies. A systematic review and meta-analysis by Li and colleagues showed the rate of severe illness is about 23% and the mortality rate is about 6%. [69] Also, individuals with risk factors such as immunodeficiency, diabetes, or old age are more
likely to have severe health outcomes than individuals without. Still, a mortality rate of 6% translates to hundreds of millions of deaths when transmission is worldwide. I include these statistics not to undermine the importance of COVID-19 or to alarm the reader but to illustrate the wide-range of symptomatologies that patients may experience.

In this discussion of COVID-19 pathophysiology I present the five categories the National Institutes of Health (NIH) uses to describe illness progression. These categories are: Asymptomatic/Presymptomatic Infection, Mild Illness, Moderate Illness, Severe Illness, and Critical Illness. An overview of each of these categories is given in Table 1.2. The progression from moderate to severe illness roughly coincides with the effects of the cytokine storm. For my project, I am focusing mainly on antivirals (which can prevent severe infection), although I discuss both antivirals and anti-inflammatory drugs for completeness.

Table 1.2: COVID-19 Disease Classification.

<table>
<thead>
<tr>
<th>Classification</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asymptomatic/Presymptomatic Infection</td>
<td>Individuals who have tested positive for SARS-CoV-2 but have not exhibited any symptoms or clinical manifestations of COVID-19</td>
</tr>
<tr>
<td>Mild Illness</td>
<td>Individuals with mild symptoms but without shortness of breath or irregular chest imaging</td>
</tr>
<tr>
<td>Moderate Illness</td>
<td>Individuals who show evidence of lower respiratory tract infection either through symptoms or radiologic imaging and who have oxygen saturation (SpO2) ≥ 94%</td>
</tr>
</tbody>
</table>
### Severe Illness

Individuals who have SpO2 ≤ 94% and ratio of partial pressure of arterial oxygen to fraction of inspired oxygen (PaO2/FiO2) < 300 mm Hg, respiratory rate > 30 breaths/min or lung infiltrates > 50%

### Critical Illness

Individuals with respiratory failure, septic shock, and/or multiple organ dysfunction

### 1.1.4 Variants

Since the start of the pandemic, the World Health Organization (WHO) and the Centers for Disease Control and Prevention (CDC) have established a classification system for emerging SARS-CoV-2 variants. Variants (or strains) are sub-species classifications which are used to identify viruses which share unique phenotypic characteristics. [71] That is, SARS-CoV-2 variants share mutations which results in similar transmissibility, pathogenicity, and other observable characteristics. Variants of Interest (VOIs) and Variants of Concern (VOCs) are two groups used to classify novel SARS-CoV-2 strains. VOIs are variants with mutations that may cause enhanced transmissibility, escape from to the host immune response, and/or resistance to current drug therapies. VOCs are strains which are known to have some or all of these characteristics.

Monitoring viral variants is critical to understanding the progression of the pandemic. Because there always exists the potential for a virus to become more infectious, cause more severe illness and/or become resistant to medications, there remains a constant need to update prevention and treatment options. In particular, this is why I am interested in exploring drug combinations. When drugs exhibit distinct mechanisms of action (MOAs), mutations which simultaneously circumvent both of MOAs are less likely to occur. Thus, viral variants are less likely to emerge.

The first VOC identified was Alpha (B.1.1.7). The Alpha variant was first recognized in
the United Kingdom in December 2020. [72, 73] Alpha has 17 mutations—eight of which are in the S protein. [71] Of these, N501Y has been shown to increase affinity of S for host ACE receptors. [74, 75] Expectedly, transmission of this VOC was enhanced compared to the Wuhan strain. [74] Perhaps the most alarming characteristic of this variant is that it is associated with an increased risk of death. In fact, several studies reported hazard ratios greater than 1.5. [76, 77, 78] Despite these troublesome figures, vaccines still provided individuals with roughly the same protection against the Alpha and Wuhan strain. [79, 80, 81, 82, 83]

Beta (B.1.351) is the second VOC and caused a wave of COVID-19 infections in South Africa in October 2020. [84] Beta has 9 mutations in the S protein, 3 of which (K417N, E484K, and N501Y) increase binding to ACE receptors. [85] Like Alpha, Beta is also associated with increased transmissibility; however, unlike Alpha this particular strain is associated with reduced protection by vaccines. [86, 87] Beta was not linked to higher mortality rates compared to the Wuhan strain but the Beta variant was linked to increased hospitalizations. [88]

The third VOC, Gamma (P.1 lineage), was identified in Brazil during December 2020. [89] Gamma has 10 mutations in the spike protein and three specifically in the Receptor Binding Domain of S (L18F, K417N, and E484K). Like Alpha and Beta, Gamma is linked to increased transmission. [74, 90, 91] Also like Beta, vaccines were shown to be less effective against preventing disease for Gamma. [92] Gamma was linked to increased hospitalizations and ICU admissions. [79, 88] However, the study by Funk and colleagues found a decreased risk of death (Adjusted Odds Ratio=0.6) for Gamma when individuals were not matched by age, sex, country, and week of reporting.

Delta (B.1.617.2) was found in India in December 2020. [71] Delta has 10 mutations in the spike protein and many of these increase binding of S to ACE2. [58, 93, 94] Indeed, evidence suggests Delta has increased transmissibility compared to the Wuhan strain. [95] Perhaps the most salient characteristic of Delta is the variant’s tenancy to cause severe illness. Delta is associated with a higher risk of death compared to any other VOC so far.
Additionally, vaccine effectiveness is reduced compared to the original strain. [82] The last VOC, Omicron (B.1.1.529) was identified in South Africa during November 2021. Omicron has upwards of 30 alterations on the spike protein. [97] These features have resulted in a 13-fold increase in viral infectivity compared to the reference strain. [98] Also, several monoclonal antibodies which were previously indicated for COVID-19 have had their emergency use authorizations (EUAs) revoked because they are no longer effective against this variant. [71] Finally, vaccine effectiveness against Omicron is also very low, with Omicron having a vaccine escape capability 14 times that of Delta. [98]

### Table 1.3: COVID-19 Variants of Concern.

<table>
<thead>
<tr>
<th>Variant</th>
<th>More Infectious</th>
<th>Higher Mortality Risk</th>
<th>Vaccine Escape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Beta</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Gamma</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Delta</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Omicron</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 1.2 SARS-CoV-2 Pharmaceuticals

Broadly speaking, pharmaceuticals against SARS-CoV-2 can be divided into prevention and treatment options. Prevention options for COVID-19 include vaccines and drug molecules. Prevention itself can be divided into two categories: pre-exposure prophylaxis (PrEP) and post-exposure prophylaxis (PEP).

PrEP describes prevention that occurs before a known exposure to SARS-CoV-2. Vaccines are currently the recommended option for PrEP although there does exist one medication, Evusheld, which is a combination of the monoclonal antibodies (mAbs) tixagevimab and cilgavimab. It should be noted that Evusheld should only be used by patients who cannot get vaccinated because they are immunocompromised (and will not mount a strong
enough immune response to vaccines) or who have a history of severe adverse reactions to COVID-19 vaccines. [70]

PEP is a strategy to prevent infection that occurs when an individual has a known exposure to SARS-CoV-2. There are currently no recommended PEP options for COVID-19 because the latest VOC, Omicron, does not respond to previous PEP options. Indeed, the NIH recommend against using bamlanivimab plus etesevimab and casirivimab plus imdevimab because these agents are no longer effective against Omicron. [70] This development underscores the continual need to develop agents which are effective against SARS-COV-2 at various stages of infection.

Treatment options for SARS-CoV-2 can also be divided into two categories. These include antivirals and immune system modulators. Antivirals are those medications which directly disrupt any step in the life cycle of the virus (see Figure 1.3). These agents halt the propagation of the virus in the body and limit future infection of cells.

Immune system modulators are medications that alter the host’s immune response. Often, tissue damage during COVID-19 occurs because the host’s immune cells mount a disproportionate attack on the host body. Immune system modulators serve to dampen or modify this response and stop future damage.

In this publication, we will discuss vaccines, antivirals, and immune system modulators because these represent the majority of options available to combat SARS-CoV-2.

1.2.1 Vaccines

In the United States, there is currently one SARS-CoV-2 vaccine authorized for emergency use and two with full FDA-approval. These are: JNJ-78436735 (Johnson & Johnson), Comirnaty (BioNTech, Pfizer), and Spikevax (Moderna). JNJ-78436735, the vaccine authorized for emergency use, is a viral vector vaccine while Comirnaty and Spikevax are mRNA vaccines. [99, 100] Although both vaccines work by having the host translate mRNA into S protein, the method by which this mRNA is produced differs between viral vector vaccines and mRNA vaccines.
JNJ-78436735 uses adenovirus 26 CoV2 to deliver DNA (which encodes for the S protein) into human cells. The viral vector releases DNA into the cytoplasm which subsequently travels the nucleus and is transcribed into mRNA. Importantly, the adenovirus vector is modified such that its DNA does not have genes which allow it to replicate. Comirnaty and Spikevax, on the other hand, use lipid nanoparticles (LNPs) that encapsulate mRNA directly. These LNPs enter the cell by endocytosis and the mRNA is delivered to the cytoplasm where it can be translated by host ribosomes.

Once cells start producing S protein, host immune cells identify the foreign antigen and mount a response. Plasma B cells begin producing antibodies which can specifically recognize portions of the S protein. This recognition allows for a rapid response from vaccinated individuals to SARS-CoV-2 following exposure. Importantly, these antibodies
serve to directly neutralize SARS-CoV-2 as well as accelerate the response from other components of the immune system. Given the complex nature of the immune response (which is outside the scope of this publication), the reader is encouraged to consult with outside resources on the topic of immunology. Here, a brief mention of antibody-mediated immunity was included because it helps the reader appreciate how vaccines function and because it illustrates how certain drugs—namely mAbs—work.

Table 1.4: Summary of COVID-19 Vaccines.

<table>
<thead>
<tr>
<th>Vaccine Manufacturer</th>
<th>Type</th>
<th>Approval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson &amp; Johnson</td>
<td>adenovirus vector</td>
<td>Emergency</td>
</tr>
<tr>
<td>Moderna</td>
<td>mRNA</td>
<td>Full</td>
</tr>
<tr>
<td>Pfizer</td>
<td>mRNA</td>
<td>Full</td>
</tr>
</tbody>
</table>

1.2.2 Antivirals

Ritonavir-boosted nirmaterlvir (Paxlovid) is a drug combination which works to inhibit the main SARS-CoV-2 protease, \( M^{\text{PRO}} \). Presently, Paxlovid is the first-line of therapy recommended for SARS-CoV-2 infection. [70] Paxlovid is an excellent introduction to the benefits of combination therapy (discussed later). The two medications comprising Paxlovid have different mechanisms of action which complement one another—reducing the risk of hospitalization or death by 89% in non-hospitalized adults. [101] Ritonavir is the agent responsible for \( M^{\text{PRO}} \) inhibition while nirmaterlvir helps increase the plasma concentration of ritonavir via Cytochrome P-450 3A4 (CYP3A4) inhibition. Because many pharmaceuticals are metabolized by CYP3A4, caution must be used by practitioners when prescribing Paxlovid.

Sotrovimab is a mAb which disrupts fusion of SARS-COV-2 with the host cell membrane. Prior to the emergence of Omicron, there were many mAbs which were recommended for the treatment and/or prevention of COVID-19. Sotrovimab is recommended for use against all strains of SARS-CoV-2 likely because its mechanism of action is not de-
dependent on the strain of SARS-CoV-2. \[102, 102\] Sotrovimab does not bind at the interface between S and ACE2—but rather, at an allosteric site. Also, the epitope to which Sotrovimab binds is highly conserved among SARS-CoV, SARS-CoV-2 and other Sarbecoviruses and is likely to remain unchanged during the evolutionary course of SARS-CoV-2. This is in contrast to the receptor-binding domain (RBD) of S which has mad many mutations since the start of the pandemic and which is susceptible to treatment escape. \[91, 103\]

Remdesivir is metabolized into a nucleotide analog of adenosine triphosphate (ATP), Remdesivir triphosphate (RDV-TP) in the host body. Replication of viral RNA by the RNA-dependent RNA polymerase (RdRP) function of nsp12 is prematurely terminated by Remdesivir, halting the viral replication cycle. Specifically, RdRP preferentially incorporates RDV-TP over ATP. This action leads to delayed-chain termination at a position 3 nucleotides ahead of RDV-TP (i+3 termination). \[104\]

Molnupiravir, like Remdesivir, targets RdRP. However, unlike Remdesivir, the mechanism of action of molnupiravir is induced mutagenesis. \[105, 106\] Molnupiravir is first metabolized inside the host to become \(\beta\)-d-N4-hydroxycytidin (NHC) triphosphate. Once in its active form, NHC triphosphate is incorporated into nascent negative-sense RNA as a Molnupiravir-like base (M) every time the template strand has an Adenosine (A) or Guanine (G) base. Next, positive-sense RNA is recreated from the negative-sense template; however, for every M, either A or G can be incorporated. Thus, every A or G replaced by M in the viral genome has a chance to become G or A, respectively. Repeating this process across the entire genome results in the production of mRNA which does not code for functional proteins and, by extension, a functional virus.

Table 1.5: Summary of COVID-19 Antivirals.

<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paxlovid</td>
<td>M(^\text{PRO})</td>
<td>Inhibition of main viral protease</td>
</tr>
<tr>
<td>Sotrovimab</td>
<td>S</td>
<td>Disruption of S binding to ACE2</td>
</tr>
<tr>
<td>Remdesivir</td>
<td>nsp12</td>
<td>Disruption of RNA-dependent RNA synthesis</td>
</tr>
</tbody>
</table>
1.2.3 Immune System Modulators

Dexamethasone is an anti-inflammatory agent in a class of medications called corticosteroids. In general, corticosteroids dampen the immune response through a variety of pathways. Although there is much overlap between how corticosteroids function, dexamethasone is likely the preferred agent because it is inexpensive, has a long half-life, and has little mineralocorticoid side effects. [107, 108, 109] The mechanisms of action of dexamethasone against COVID-19 are many and can be divided into two large groups: genomic mechanisms and non-genomic mechanisms. [110, 107, 109, 111] Typically genomic mechanism are observed even at low concentrations of drug so these will be discussed in this section. [107] Dexamethasone readily diffuses across cellular membranes due to its lipophilic character. Once in the cytoplasm, Dexamethasone can bind to glucocorticoid receptors (GRs) to initiate their translocation into the nucleus. Once in the nucleus, GR can exert its genomic anti-inflammatory mechanisms. In short, activated GR induces the transcription of anti-inflammatory genes and repress transcription of pro-inflammatory genes—many of which are related to the cytokines which cause the infamous cytokine storm. [110]

Baricitinib and tofacitinib belong to a class of drugs called Janus Kinase (JAK) inhibitors. JAKs are proteins which phosphorylate Signal Transducer and Activator of Transcription (STAT) proteins. Humans have four JAK (JAK1, JAK2, JAK3, and TyK2) and seven STAT (STAT1, STAT2, STAT3, STAT4, STAT5A, STAT5B, and STAT6) proteins. [112] Once, phosphorylated by JAKs, STATs dimerize and translocate to the nucleus where they activate transcription of various genes—many of which are cytokines. [113, 114, 115] Thus, like corticosteroids, JAK inhibitors also exert their anti-COVID-19 effects by reducing the immune response at the level of transcription. It should be noted that JAK inhibitors are used in conjunction with dexamethasone—not as standalone treatment—because dexamethasone is seen as the standard of care in patients with severe COVID-19.
Tocilizumab and sarilumab are monoclonal antibodies which bind to both soluble and membrane-bound interleukin-6 (IL-6) receptors. IL-6 is an inflammatory cytokine produced by many cell types including lymphocytes (B cells and T cell), monocytes, fibroblasts, and bronchial epithelial cells. IL-6, has been known to be an indicator of various kinds of cytokine storms (including the one brought about by SARS-CoV-2). Indeed, IL-6 is known to participate in pathways that increase vascular permeability and leakage—two processes known to contribute to the pathophysiology of pulmonary dysfunction in severe COVID-19 cases. With this in mind, the reader can appreciate why these agents are sometimes prescribed to COVID-19 patients in addition to dexamethasone.

Table 1.6: Summary of COVID-19 Immune System Modulators.

<table>
<thead>
<tr>
<th>Name</th>
<th>Target</th>
<th>Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dexamethasone</td>
<td>Glucocorticoid Receptors</td>
<td>Transcription regulation</td>
</tr>
<tr>
<td>Baricitinib</td>
<td>Janus Kinase Receptors</td>
<td>Transcription regulation</td>
</tr>
<tr>
<td>Tofacitinib</td>
<td>Janus Kinase Receptors</td>
<td>Transcription regulation</td>
</tr>
<tr>
<td>Tocilizumab</td>
<td>Interleukin-6 Receptors</td>
<td>Reduction of IL-6 induced inflammation</td>
</tr>
<tr>
<td>Sarilumab</td>
<td>Interleukin-6 Receptors</td>
<td>Reduction of IL-6 induced inflammation</td>
</tr>
</tbody>
</table>
Chapter 2

Background

2.1 Molecular Descriptors

Molecular descriptors encapsulate information about a molecule—usually a small molecule (<900 u)—in a computer-usable format so that computations may be made. Molecular descriptors may output scalars, vectors, 2D arrays (matrices), or even higher-dimensional arrays; however, for our purposes, all molecular descriptors transform molecules into vectors of a fixed size. Here, I give a brief overview of molecular descriptors and a more detailed account of the descriptors used in my work.

Molecular descriptors may be grouped into categories based on their dimensionality. [122, 123, 124, 125, 126] Note, this dimensionality is distinct from the output dimension described in the previous paragraph and instead refers to the method by which the molecular descriptor is calculated. In the scientific community, there exists various—sometimes conflicting—definitions of this dimensionality. Thus, the following definitions will serve to classify the molecular descriptors as they are used in this publication.

0D: Zero-dimensional molecular descriptors are "bulk" properties of molecules obtained by considering atoms individually. They are often relatively easy to calculate because they require no knowledge of the connectivity of a molecule—hence, zero-dimensional. Some examples include atom counts for a particular atom, molecular weight, or sums of the atomic radii.

1D: One-dimensional molecular descriptors are calculated by considering fragments of
molecules. Because molecules can be represented by mathematical graphs, descriptors that are calculated by traversing an entire molecule are termed two-dimensional. Naturally, descriptors calculated by traversing only part of a molecule, are termed one-dimensional. Counts of functional groups such as ketone, nitrile, or aryl groups are examples of one-dimensional features.

2D: Some descriptors require traversing or considering all of the atoms in a molecule. Descriptors which are calculated in this manner are second-dimensional. These descriptors encode information about the connectivity of a molecule as it is represented in its corresponding structural formula. An important remark is that these descriptors are not sensitive to the arrangement of atoms in 3D space and, consequently, information such as stereochemistry is lost on these features. Examples of 2D features include Extended-Connectivity Fingerprints (ECFPs) [127], Morgan Fingerprints, and Chemically Advanced Template Search (CATS).

3D and Higher: Molecules are, of course, three-dimensional objects. As such, molecular fingerprints which account for the structural nuances inherent in the three-dimensional arrangement of atoms are better poised to describe molecules interacting with biological systems. [128, 129, 130, 131] This advantage, however, comes at a cost. Molecules can exhibit various conformations in nature. Choosing the right conformer/s for a particular application requires choosing an optimization strategy for selecting a suitable arrangement of atoms. Thus, three-dimensional fingerprints are dependent on the chosen optimization strategy. To make matters worse, molecules are not static but dynamic entities which may adopt many different conformers within a given time and biological context. Though they are invaluable tools for drug discovery, three-dimensional descriptors require careful application and are not used in these studies. There exists higher dimensional molecular descriptors, but the discussion of these is beyond the scope of this publication.
2.1.1 Molecular Descriptors Used

2.1.1.1 RDKit Descriptors

RDKit has hundreds of 0D, 1D, 2D, and 3D descriptors available for use. A complete list of descriptors can be found on the RDKit documentation website. For my work, RDKit Descriptors includes all those given by running the following code:

```python
from rdkit.Chem import Descriptors

desc_list=[x[0] for x in Descriptors._descList]
print(desc_list)
```

Figure 2.1: Cartoon Representation of a Molecular Descriptor. [132]
The full list of descriptors is given in section 6.1. Using these descriptors produces a vector of length 208. A brief overview of families of descriptors is given in Section 6.2.

2.1.1.2 Extended Connectivity Fingerprints

Molecular fingerprints are a special type of descriptor that encode information about pieces of a molecule in a vector of fixed size. This vector can encode how many times a substructure is found in a molecule (a count vector) or the presence of a substructure in general (a bit vector). Numerous fingerprints have been developed. For my work, I use an Extended Connectivity Fingerprint (ECFP) because this family of fingerprints has been extensively studied and benchmarked. I will not elaborate on the ECFP algorithm itself; instead, I will focus on select aspects of ECFPs so the reader can understand why the fingerprint is used in this study. The reader is, of course, encouraged to supplement their knowledge by consulting the literature. [127]

I begin my discussion of ECFPs by providing a motivation for their use. The list of RDkit descriptors in Section 6.1, though computationally easy to calculate, suffers from a problem called degeneracy. That is, molecules will have similar RDkit descriptors even thought they can exhibit different behaviors biologically. Indeed, many of the descriptors in the list are 1D or even 0D so they do not make use of all of the structural information that a molecule can provide. Because the problem at hand is to classify combinations of molecules according to their capacity for synergy, ideally we would want descriptors which could readily differentiate molecules.

Enter ECFPs, which are a computationally inexpensive way to capture much of the structural information present in a molecule. ECFPs are an application of the Morgan Algorithm which is used to determine if two molecules, with different atom numberings schemes, are identical. [133] Essentially, ECFPs require the user to apply a variation of the Morgan Algorithm but stop after a pre-determined number of iterations. This number, called the radius, identifies the particular ECFP which is used. When writing a particular ECFP, the diameter (not the radius) is reported. For example, ECFP_4 is the ECFP
algorithm with a radius of 2. The ECFP algorithm begins by considering every heavy atom in a molecule and the bonds it forms to other heavy atoms—these are the features at iteration 0. After the first iteration, the radius is increased by one—that is, the neighbors of each of features are appended to the original. This process continues until the user-specified radius is achieved. Each of the features is then encoded into a bit or count vector of a fixed size. Often only 2 iterations are needed for applications such as clustering. [127] This observation is why, for my work, I did not use an ECFP with a large radius, and instead decided that ECFP_6 would suffice.

### 2.2 Machine Learning

In short, machine learning (ML) is an approach whereby an algorithm attempts to use data to improve its performance on a particular task. ML has historically been divided into three categories: supervised learning, unsupervised learning, and reinforcement learning. In this particular project, I exclusively use supervised learning.

Supervised learning attempts to make use of labeled data to make predictions. User-provided inputs and user-provided outputs (labels) are fitted to make models. The idea behind supervised machine learning is that with enough quality data, an algorithm can learn the inputs that lead to certain outputs.

A supervised machine learning model can further be understood in terms of the problem it is attempting to solve. Classification problems attempt to use inputs to place a sample into certain categories. Regression problems, instead, output certain numerical values (commonly real numbers). In this background section, I manufacture a small example so the reader can get a sense of the machine learning workflow. In my example, I will attempt to classify a sample as a dog, cat or snake.

I begin by dividing the basic supervised learning workflow into the following seven steps:

1. **Find Labeled Data**
The first step is to find labeled data. In my example (Figure 2.2) I simply generated six different samples—in this case, animals. Note: all images are stock images obtained from Microsoft PowerPoint version 2203. The data are labelled because we know what the animal is *a priori*.

![Figure 2.2: Labeled Animal Data.](image)

2. **Decide Important Features**

The next step is to decide the features might be most important for a model. Suppose we have tabular data on the animals in step 1. We can choose the features which we might want to train the model on. In this example, I choose the weight of the animal, the noise the animal makes, and the type of ears the animal has. There could be many more features (like length, fur/scales, etc.) that I did not choose, however. In practice, there often exists features to choose from for a given problem and often one considers only a subset of available features.

In this example, notice that noise is a very informative feature because only cats meow, dogs bark and snakes hiss. Also, notice that weight is not very informative because there can be an overlap between the weights of cats, dogs, and snakes. The
Table 2.1: Important Features from Animal Data.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Weight (lbs)</th>
<th>Noise</th>
<th>Ears</th>
<th>Animal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8.2</td>
<td>Meow</td>
<td>Pointy</td>
<td>Cat</td>
</tr>
<tr>
<td>2</td>
<td>15.1</td>
<td>Bark</td>
<td>Floppy</td>
<td>Dog</td>
</tr>
<tr>
<td>3</td>
<td>8.0</td>
<td>Bark</td>
<td>Pointy</td>
<td>Dog</td>
</tr>
<tr>
<td>4</td>
<td>6.2</td>
<td>Hiss</td>
<td>Hole</td>
<td>Snake</td>
</tr>
<tr>
<td>5</td>
<td>7.9</td>
<td>Hiss</td>
<td>Hole</td>
<td>Snake</td>
</tr>
<tr>
<td>6</td>
<td>7.8</td>
<td>Meow</td>
<td>Pointy</td>
<td>Cat</td>
</tr>
</tbody>
</table>

process of choosing only certain features based on their anticipated benefit to the machine learning model is called feature selection.

3. Make Data Machine Learning Ready

Data need to be in matrix form for machine learning. Rows represent samples and columns represent features. The first step to achieve matrix form is to encode categorical data into integers. In this example the encoding is as follows: meow=1, bark=2, hiss=3; pointy=1, floppy=2, hole=3; cat=1, dog=2, snake=3. Often-non categorical data (like weight) may be normalized through different methods. Normalization ensures that data fall within a specified range. In this example, we forgo this step. Also, in some machine learning workflows, normalization occurs after the the data set is split (see step 4).

The processed data can be arranged in an augmented matrix, $D$, where the last column represents the category. Alternatively, the matrix $D$ can be split into an input matrix, $X$, and an output vector, $y$. For simplicity I choose the augmented matrix approach.
4. **Split Data**

To ensure a model can be evaluated, one must partition data into training and testing sets. The training data are used to develop a model and the testing (holdout) data are used to gauge model performance. In this example we use a 50/50 train/test split which means that 50% of data are used for training and 50% are used for testing. A more common scheme, however, is a 70/30 train/test split. There is no pair of numbers which are best for a split; however, one must ensure that there is enough data for a model to learn from the training set.
5. **Train Model**

Once data are split, training data are fed into a specific machine learning algorithm. A discussion of the algorithms used in my work is given in Section 2.2.1. An important note is that sometimes a subset of training data is used specifically for hyperparameter optimization. See Section 2.2.2 for more information.

6. **Test Model**

After a model has been developed, predictions are made using the test set. The test inputs are introduced into the model and an output vector, $\hat{y}$, is produced.

7. **Evaluate Model**

Lastly, the test split portion of $y$ (the known test outputs) and $\hat{y}$ (the predicted test outputs) are compared. An analysis using various metrics provides insight into model performance. For more information about machine learning metrics, please read Section 2.2.3.
2.2.1 Algorithms Used

In the following section, I give a brief description of the machine learning algorithms used in my work. In particular, I focus on classification although both Random Forest and Multi-layer Perceptron can make regression predictions.

2.2.1.1 Random Forest (RF)

Random Forest (RF) is an ensemble algorithm that classify a sample or predict a regression value. The main idea behind RF is to use many different decision trees (randomly created) to arrive at a predicted value. For classification, these individual decision trees each cast a vote for a particular category and the category with the most votes is the predicted result. To give the reader a sense of how the algorithm operates, I first explain how to create decision trees and then how to combine them in a RF.

Decision trees are graphs with one root node and two or more leaves (terminal nodes). Every non-leaf node represents a binary decision where one feature is used to split samples. Every split should represent progress toward classification whereby the two groups of the split contain a purer collection of samples. A measure of purity can take many mathematical forms, but the one I used in my project is Gini (Equation 2.1). Figure 2.3 shows how the process of minimizing Gini occurs for three different decision trees. At the root of the tree, all samples are light green and have a Gini of 1 because there are just as many green samples as white samples. As samples travel down the decision tree, the nodes become purer until they achieve a Gini of 0—that is, until the leaves have only green samples or
Given the same data and user-provided parameters, a decision tree will always be built in the same way. This explains why decision trees are prone to learn the nuances of the data used to generate them and why they fail at classifying samples from new datasets. To get around this, RF uses two techniques: bootstrapping and selecting a subset of features. Bootstrapping means randomly sampling with replacement from the original dataset. Often, RF samples the same number of times as there are entries in the dataset. This means that every tree built by RF is trained on different data. Also, only a portion of the features is used to make the trees. Commonly, a number approximately equal to the square root of total features is used for each tree. These features are, again, randomly selected. Together, these two techniques help build different trees in the forest. Once a user-defined number of trees is built, the RF can take new data that is passed to individual individual trees for voting. Classification for the sample is determined by the greatest number of votes.

\[
Gini(t) = 1 - \sum_{i=1}^{K} [p(i|t)]^2
\]

where K represents the number of classes, and p(i|t) is the proportion of items belonging to class i at node t.
2.2.1.2 Multi-layer Perceptron (MLP)

A neural network is a machine learning algorithm which uses neurons (perceptrons) as information passing units. The neural network used in my project is the Multi-layer Perceptron (MLP). MLPs have an input layer, one or more hidden layers (depending on the design of the user), and an output layer. The numerical data contained in a sample is passed first through an input layer. This input layer has as many neurons as there are features in the dataset and each neuron receives as input the corresponding value of the feature. For example, if we have four features, then the input layer can be represented by $n1, n2, n3, n4$ on the left side of Figure 2.4. Next, at the first hidden layer—as well as every other hidden layer—there is a user-defined number of neurons. Every neuron in the input layer has a connection to every neuron in the first hidden layer. Similarly, every neuron in the first hidden layer has a connection to every neuron in the second hidden layer (if a second hidden layer exists). Finally, for classification problems the output is a vector
of probabilities for each class where the sum of the individual components of the vector must add up to 1. In Figure 2.4 we have a binary classification problem because there are two neurons in the output layer.

The learning that occurs in MLPs happens at the connections between neurons. If we design our MLP as it is in Figure 2.4 then we can describe the math that allows MLPs to learn. At every layer except the input layer, all of the nodes on the right receive input from all of the nodes on the left with some modifications. These modifications are determined by a weighted sum, a bias, and a user-specified activation function. For example, we will consider the n12 node. This node receives a weighted sum of inputs from all nodes of the previous layer (represented by the four arrows pointing to n12). A bias is then added to this weighted sum. The result is then passed through an activation function. The activation function is a hyperparameter that can be set by the user. The action of these three components determines the input to n12. This process is repeated for all the nodes in the hidden layers and the output layer as information is propagated left to right.

Training an MLP involves finding values for all the weights and biases of the network. These values should be chosen such that the neural network can correctly classify different inputs. For a given input/output pair, a cost function is used quantify how well the weights and biases of a neural network produce a result close to the true output. Thus, tuning the weights and biases of a neural network requires minimizing the cost function. While there are many different costs functions and many different minimization techniques, for my work, I used the scikit-learn implementation of the squared error function and the "Adam" stochastic gradient descent method to minimize this cost function. [134] Although the specifics of this approach are beyond the scope of this publication, the reader is encouraged to read more about these tools to better understand MLPs.
2.2.2 Hyperparameter Optimization

Hyperparameters are user-set components of machine learning algorithms which can affect model performance. Available hyperparameters depend on the chosen ML algorithm. Ideally, hyperparameters are adjusted (tuned) so that they maximize model performance but allow the model to generalize to non-training datasets. In this section, I cover the hyperparameters I tuned for RF and MLP and the strategies used to adjust these.

2.2.2.1 RF Hyperparameters

Number of Estimators:
Number of Estimators controls the number of trees to use in RF.

Maximum Depth:
Maximum depths controls the maximum number of splits that any one sample might encounter in a decision tree.
Minimum Samples Split:
Minimum Samples Split is a lower bound for the number of samples required to split an interior (non-root and non-leaf) node. Nodes having at least this number will be split.

Minimum Samples Per Leaf:
Minimum Samples Per Leaf is a lower bound for the number of samples required to be at a leaf. Splits from a node to a leaf will only be considered if they leave at least this number of samples in both the left and right leaves.

2.2.2.2 MLP Hyperparameters

Hidden Layer Sizes:
Hidden Layer Sizes is a tuple value which determines the number of neurons in each hidden layer. The tuple (100,50) represents a model with 100 neurons in the first hidden layer and 50 in the second.

Activation Function:
The activation function is used to scale an input into a neuron. Typically, activation functions have behaviors which compress values to a desired range.

Alpha:
Alpha is a regularization parameter which determines the penalty for overfitting data.

2.2.2.3 Grid Search

Grid Search is a technique whereby all possible combinations of hyperparameters in a user-provided grid are tested. For my project, I used the hyperparameter grids given by Tables 2.6 and 2.7. If we consider Table 2.6, then $3 \times 4 \times 3 \times 3 = 108$ combinations were tested. Similarly, 27 combinations were tested for Table 2.7. For all combinations, a particular scoring metric is chosen so that the combinations can be compared. For example, for a classification problem, Grid Search might try to optimize for accuracy. If this is the case, the combination of hyperparameters which maximizes accuracy is chosen for the model.
Table 2.6: Random Forest Hyperparameter Grid.

<table>
<thead>
<tr>
<th>Number of Estimators</th>
<th>Max Depth</th>
<th>Minimum Samples Split</th>
<th>Minimum Samples Per Leaf</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>None</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>200</td>
<td>8</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.7: Multi-layer Perceptron Hyperparameter Grid.

<table>
<thead>
<tr>
<th>Hidden Layer Sizes</th>
<th>Activation Function</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>(100)</td>
<td>Rectified Linear Unit (ReLU)</td>
<td>0.0001</td>
</tr>
<tr>
<td>(100,50)</td>
<td>logistic</td>
<td>0.005</td>
</tr>
<tr>
<td>(100,100)</td>
<td>hyperbolic tangent (tanh)</td>
<td>0.05</td>
</tr>
</tbody>
</table>

2.2.2.4 K-Fold Cross Validation

Tuning hyperparameters requires model performance be gauged before evaluating the model on the test set. One solution is to partition the training set into a sub-training set and a validation set. A model with chosen hyperparameters is then trained on the sub-training set and the performance is gauged with the validation set. This approach can be repeated for many different sets of hyperparameters until an optimal set is found. Although straightforward, this technique requires the sub-training and validation set be large enough to allow for effective hyperparameter tuning.

Often, when there is limited training data to begin with another approach called K-Fold Cross Validation is used. The training data is divided into K-Folds. Then, K sub-training datasets are formed by withholding a different fold each time. The model is training on these sub-training datasets and evaluated with the corresponding test fold. A scoring metric (the same one mentioned in Section 2.2.2.3) is averaged using the K evaluations. The best value for this average will be associated with an optimal set of hyperparameters.
The resulting model with the optimal hyperparameters is then evaluated by the test set.

2.2.3 Evaluation Metrics

Depending on the problem a model is trying to solve, ML metrics can be either classification-based or regression-based. Here I discuss classification metrics because these are used to evaluate my model.

An additional remark: some classification metrics (including Recall, Precision, and F1 score) can be reported on a per-class basis or as an average value for all classes. Though there are different methods for averaging, this analysis these three metrics are averaged using the weighted method. The weighed method calculates the metric for each class, scales this value by the number of correctly identified samples, adds all of the scaled values, and then finally divides by the total number of samples. The weighted method is used in this analysis because it is resistant to class imbalance.
Confusion Matrix

A confusion matrix summarizes prediction results for the classification problem. For a multi-class problem, the matrix is $\mathbb{N}^{m \times m}$, where $m$ represents the number of classes. The rows represent true labels and the columns represent predicted labels. For my work, $m=3$ and these categories are antagonistic, additive, and synergistic. An example confusion matrix, $A$, with $n=50$ samples (manufactured data) is given in Figure 2.8. This confusion matrix will be used for demonstration purposes to calculate classification metrics. Indexing the confusion matrix will be done by $M_{i,j}$ where $i$ and $j$ are the row number and column number, respectively, and the top left entry of the confusion matrix is $A_{1,1}$.

Importantly, from the confusion matrix, one can ascertain True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values for each class. Formally, these are found with the following definitions:

$$TP = A_{i,i} \quad (2.2)$$

$$TN = \sum_{q \neq i} \sum_{j \neq i} A_{q,j} \quad (2.3)$$

$$FP = \sum_{j \neq i} A_{j,i} \quad (2.4)$$

$$FN = \sum_{j \neq i} A_{i,j} \quad (2.5)$$

where $i,$ represents the index of the chosen class label.

Let us consider the antagonistic class as an example. This class label has an index of 1. The TP values are found in the cell with row and column number 1. In this case, the value is 15. The TN values are found by all row/column pairs for which neither the row nor column are antagonistic. This is $12 + 2 + 2 + 13 = 29$. The FP values are those with the antagonistic column and either the additive or synergistic row. Here, we have $1 + 0 = 1$. Lastly, the FN values are those with the antagonistic row but additive or synergistic columns. Specifically,
Table 2.8: Example Confusion Matrix.

<table>
<thead>
<tr>
<th></th>
<th>Antagonistic</th>
<th>Additive</th>
<th>Synergistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antagonistic</td>
<td>15</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Additive</td>
<td>1</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>Synergistic</td>
<td>0</td>
<td>2</td>
<td>13</td>
</tr>
</tbody>
</table>

this is 2+3=5. We will use these values in our example calculations for recall, specificity, precision, and F1-score.

**Accuracy**

Accuracy is defined as the sum of correctly predicted samples divided by the total number of samples. This value is not calculated on a per-class basis. In terms of Table 2.8 this requires us to sum all of the diagonal elements and divide by \( n \).

\[
\text{Accuracy} = \frac{\sum A_{i,i}}{n} = \frac{40}{50} = 0.8 \tag{2.6}
\]

**Recall**

Recall is the number of true positive samples over total positive samples. High recall indicates that a model is good at identifying a particular label—even if that means sometimes having false positives. Here we report the recall for the antagonistic class.

\[
\text{Recall} = \frac{TP}{TP+FN} = \frac{15}{15 + 5} = 0.75 \tag{2.7}
\]

**Precision**

Precision is the number of true positive samples over samples predicted positive. High precision means that a model does not incorrectly predict a particular label—even if that means sometimes having false negatives. Here we report the precision for the antagonistic class.

\[
\text{Precision} = \frac{TP}{TP+FP} = \frac{15}{15 + 1} = 0.94 \tag{2.8}
\]

**F1-Score**

From the descriptions of recall and precision above, we can observe that these two metrics
are often at odds. A good model should balance false positive and false negative errors. To quantify this, F1-Score is the harmonic mean of these two values. Here we report the F1-Score for the antagonistic class.

\[
F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 0.83
\] (2.9)

**Weighted Average**

Weighed average is a method to combine a classification metric among different class labels.

\[
\text{Weighted Average} = \frac{1}{n} \sum_{i} TP_i M_i
\] (2.10)

where \(i\) represents a class label, \(TP_i\) represents the TP value for class label \(i\), and \(M_i\) represents the value of a chosen metric (recall, precision, or F1-Score) for class label \(i\).

### 2.3 Synergy

Colloquially, synergy is the unexpected benefit that occurs when two or more items are combined. In the realm of drug discovery, this definition can be applied to combining two or more drugs and observing the resulting therapeutic effects (TEs). There are many different strategies for quantifying synergy and a brief discussion of two measures will be presented. The first of these measures, Bliss Independence, takes as input a matrix of TEs for drug combinations and outputs a matrix of synergy values (\(\Delta\text{Bliss}\)). In our case, TEs are represented by percent cell survival after infection with SARS-CoV-2.

Often, synergy is a dose-dependent phenomenon. For instance, at certain concentrations Drug X and Drug Y are synergistic but at different concentrations they are antagonistic. In this way, a synergy matrix helps illuminate the so-called synergy landscape for drug combinations at various concentrations. In my work, I am interested in summarizing the synergy landscape into one single scalar value to simplify model development and help answer the classification question. More precisely, I hope to use one metric to gauge whether drugs are synergistic or antagonistic. To this end, I use a measure called \(\beta\) which summarizes
Table 2.9: Example Dose Response Matrix.

<table>
<thead>
<tr>
<th></th>
<th>0.51</th>
<th>0.61</th>
<th>0.94</th>
<th>0.88</th>
<th>0.95</th>
<th>1</th>
<th>10 μM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.51</td>
<td>0.57</td>
<td>0.92</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5 μM</td>
</tr>
<tr>
<td>0.31</td>
<td>0.85</td>
<td>0.97</td>
<td>0.93</td>
<td>0.99</td>
<td>1</td>
<td>1</td>
<td>3.3 μM</td>
</tr>
<tr>
<td>0.26</td>
<td>0.42</td>
<td>0.99</td>
<td>1</td>
<td>0.99</td>
<td>1</td>
<td>1</td>
<td>1 μM</td>
</tr>
<tr>
<td>0.35</td>
<td>0.85</td>
<td>0.94</td>
<td>1</td>
<td>1</td>
<td>0.95</td>
<td>1</td>
<td>0.1 μM</td>
</tr>
<tr>
<td>0.48</td>
<td>0.88</td>
<td>0.97</td>
<td>0.92</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0 μM</td>
</tr>
<tr>
<td>10 μM</td>
<td>5 μM</td>
<td>3.3 μM</td>
<td>1 μM</td>
<td>0.1 μM</td>
<td>0 μM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

the synergy matrix produced by the Bliss Independence model. For the remainder of this section, I give a brief introduction to Bliss and β and use an example matrix to calculate these values.

2.3.1 Bliss Independence

Bliss Independence, or simply Bliss, is a measure forecasting the expected therapeutic effect of two drugs. [135] This therapeutic effect (usually expressed as percent cell survival) should be converted to a value between 0 and 1 to allow for manipulation under probability theory. From probability theory, the union of two probabilities can be expressed as the sum of the individual probabilities minus their intersection (Equation 2.11). The key assumption Bliss makes is that drugs act through distinct biological processes—hence their actions are independent and the intersection of the two probabilities can be expressed as a product (Equation 2.12). [136, 137, 138] Thus, the expected TE effect for combinations of two drugs can be predicted using simply by knowing the individual effects of the drugs at the required concentrations (Equation 2.13).

However, because the mechanism of action for drugs is not often known a priori—including for drug repurposing studies—the assumptions made by the Bliss model are not always valid. [136, 137, 138] Even so, Bliss can be a useful tool to compare the
predicted activity of a combination against the empirical activity. This difference, called ΔBliss is calculated by subtracting empirical TE from the Bliss-estimated TE (Equation 2.14). Importantly, more negative values for ΔBliss indicate greater synergy at a particular concentration combination.

\[
P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{2.11}
\]

\[
P(A \cap B) = P(A) \times P(B) \tag{2.12}
\]

\[
Bliss = E_x + E_y - E_x \times E_y \tag{2.13}
\]

where \(E_x\) and \(E_y\) are the individual TEs for drug X at concentration x and drug Y at concentration y, respectively.

\[
\Delta Bliss = Bliss - E_{xy} \tag{2.14}
\]

where \(E_{xy}\) is the TE when drug X and drug Y are combined at concentrations x and y, respectively.
Table 2.10: Example $\Delta$Bliss Matrix.

<table>
<thead>
<tr>
<th></th>
<th>0.03</th>
<th>-0.27</th>
<th>-0.03</th>
<th>-0.04</th>
<th>-0.05</th>
<th>0</th>
<th>10 µM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>-0.31</td>
<td>-0.05</td>
<td>0.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5 µM</td>
</tr>
<tr>
<td>-0.17</td>
<td>-0.03</td>
<td>0</td>
<td>0.01</td>
<td>-0.01</td>
<td>0</td>
<td>0</td>
<td>3.3 µM</td>
</tr>
<tr>
<td>-0.22</td>
<td>-0.46</td>
<td>0.02</td>
<td>0.08</td>
<td>-0.01</td>
<td>0</td>
<td>0</td>
<td>1 µM</td>
</tr>
<tr>
<td>-0.106</td>
<td>0.014</td>
<td>0.019</td>
<td>0.126</td>
<td>0.05</td>
<td>0</td>
<td>0</td>
<td>0.1 µM</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10 µM</td>
<td>5 µM</td>
<td>3.3 µM</td>
<td>1 µM</td>
<td>0.1 µM</td>
<td>0 µM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.3.2 Cokol $\beta$

Work done by Cokol et al. conveniently uses ideas from the Bliss model to convert a matrix of TEs into a value between zero and two. [139] All the entries in the matrix are summed according to Expression 2.15 with $\beta$ chosen so that it minimizes the summation. The metric $\beta$ quantifies how much the empirical TEs (the term to the left of the minus sign) deviates from the null (Bliss-predicted) synergy landscape. Notice how the terms to the right of the minus sign assume the two drugs exert their effects independently. Importantly, the original work done by Cokol et al did not consider cell survival (but rather, cell death) because the goal was to limit fungal growth. For our purposes, this simply requires that we take the difference between 1 and the probability of cell survival for each concentration combination to frame the problem in terms of cell death. Making this adjustment, we apply the equation above and arrive at $\beta=0.95$ for our example data.

$$\sum [E_{xy} - \beta \times (E_x \times E_y)]^2$$  \hspace{3cm} (2.15)

where $E_x$ and $E_y$ are the individual TEs for drug X at concentration x and drug Y at concentration y and $E_{xy}$ is the TE when drug X and drug Y are combined at concentrations x and y.
Chapter 3

Methodology

3.1 Data Preparation

All data were obtained from the National Center for Advancing Translational Sciences (NCATS) Open Data Portal. The assay type is SARS-CoV-2 Cytopathic Effect (CPE)—specifically, Drug Combination Screening Data. For specific assay information, the reader is encouraged to consult the Open Data Portal.

In total, there were two datasets. The first dataset had three batches, and the second dataset had 4 batches. For each of these batches, the drug names, NIH Chemical Genomics Center (NCGC) ID, and the synergy metric $\beta$ were extracted for every experiment (matrix). Within a batch, $\beta$ was averaged for replicates. Having extracted drugs names and $\beta$ for each batch, the results were combined into one dataset. Notably, repeated $\beta$ values for identical drug combinations were averaged across different batches.

SMILES representations were manually obtained from drug names. These SMILES strings were de-salted, canonicalized, and protonated at physiological pH using Open Babel.

3.2 Molecular Descriptors

All processed SMILES were featurized using RDKit. ECFP_6 and RDKit Descriptors were concatenated as shown in Figure 3.1. For each combination, two sets of features vectors were created. The first represent the vector in the Drug A - Drug B order (forward order) and the second in the Drug B - Drug A order (reverse order). This was done because ML
models should output the same prediction regardless of the ordering of drugs in the features vector.

Figure 3.1: Training Feature Vector Pairs.

3.3 Machine Learning

Having created features for each combination, 75% were split into a train set and 25% into a test set. Importantly, for forward and reverse order pairs, both were kept in the same (train or test) set.

Following the split, the train set was used to fit RF and MLP models using the features
vectors as inputs and $\beta$ values as outputs. Grid Search and 3-fold cross validation were used to uncover optimal hyperparameters by optimizing for root mean squared error (a regression metric).

For the test set, features vectors were introduced into RF and MLP models to obtain predicted values for $\beta$. These values were then averaged (see Figure 3.2). The averaged regression results were then converted into classification results according to the thresholds set in Table 3.1. Our classification problem is a regression problem at the back end precisely because classification results cannot be averaged.

ML models were evaluated after all $\beta$ values were converted to classification labels. The models were evaluated both on the test set and on an external dataset. The external data was taken from the work done by Jin et al. and Riva et al. [140, 141] Only combinations which were not present in the original train and test set were kept from this external set.

After evaluating the models, I used them to predict synergistic combinations from a large scale screening done in collaboration with Dr. Colleen Jonsson from The University of Tennessee Health Science Center. The screening was of 2861 compounds from Selleckchem’s FDA-Approved Drugs library. Three criteria were used to select top combinations:

1. Drugs should exhibit $> 50\%$ CPE from the screening

2. Drugs should exhibit $< 10\%$ standard deviation from the screening

3. Combinations should exhibit a predicted $\beta < 0.9$ from the ML models

The top 20 scoring combinations (those with the lowest $\beta$) were chosen for *in vitro* testing.
3.4 Combination Assays

To the extent possible, in vitro testing was done in a manner resembling the CPE assay mentioned in Section 3.1. Dr. Llano from the Biological Sciences department at the University of Texas at El Paso performed the combination assays. Specifically, thirty-six wells of SARS-CoV-2 infected Vero E6 cells were tested at various concentration combinations for each drug pair. After 72 hours of exposure to SARS-CoV-2, a luminescence reading was obtained to quantify cell survival. Luminescence values were normalized according to a positive control (cells not infected with SARS-CoV-2) and a negative control (cells infected...
with SARS-CoV-2 but given no drugs). The result is a matrix which represents percent cell survival at specific drug concentrations.
Chapter 4

Results and Discussion

4.1 Machine Learning Models

From Figure 4.1, we can see there is no statistically significant difference between RF and MLP models for any of the reported metrics as determined from the test set. Accuracy for both RF and MLP was above 0.7, indicating the models were correctly able to identify synergism, additivity, and antagonism in about 70% of all samples. Recall was also above 70% suggesting that the models had satisfactory positive-predictive ability. Recall was slightly lower than precision but still above 0.6, meaning the models were able to avoid false positive predictions for a majority of test samples but type 1 errors did occur. Lastly, F1-Score was above 0.6 and (expectedly) between Recall and Precision meaning that the models balanced false-positive, and false-negative error types in their predictions.

Having established the models’ performances on the test set, I wanted to test a model on an external dataset. I chose MLP because it performed better on the test set, though this result was not statistically significant. From Figure 4.2, one can see MLP had higher metrics for the external dataset than the test set. Specifically, accuracy, recall, and F1 Score were all above 0.7 and precision was above 0.75. With these results in mind, we used MLP to predict combinations of drugs that might be synergistic. Our top 20 predictions are shown in Table 4.1.
Table 4.1: Top Scoring Combinations.

<table>
<thead>
<tr>
<th>Drug X</th>
<th>Drug Y</th>
<th>Predicted $\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>nelfinavir</td>
<td>dronedarone</td>
<td>0.81</td>
</tr>
<tr>
<td>terconazole</td>
<td>olmutinib</td>
<td>0.81</td>
</tr>
<tr>
<td>amodiaquine</td>
<td>prochlorperazine</td>
<td>0.81</td>
</tr>
<tr>
<td>amodiaquine</td>
<td>crizotinib</td>
<td>0.82</td>
</tr>
<tr>
<td>mehydrolin</td>
<td>melitracen</td>
<td>0.82</td>
</tr>
<tr>
<td>terconazole</td>
<td>harringtonine</td>
<td>0.82</td>
</tr>
<tr>
<td>crizotinib</td>
<td>thiothixene</td>
<td>0.82</td>
</tr>
<tr>
<td>abemaciclib</td>
<td>dronedarone</td>
<td>0.82</td>
</tr>
<tr>
<td>pimavanserin</td>
<td>thiothixene</td>
<td>0.82</td>
</tr>
<tr>
<td>melitracen</td>
<td>fingolimod</td>
<td>0.82</td>
</tr>
<tr>
<td>perphenazine</td>
<td>revaprazan</td>
<td>0.82</td>
</tr>
<tr>
<td>fluphenazine</td>
<td>pimavanserin</td>
<td>0.82</td>
</tr>
<tr>
<td>tilorone</td>
<td>crizotinib</td>
<td>0.82</td>
</tr>
<tr>
<td>nilotinib</td>
<td>harringtonine</td>
<td>0.83</td>
</tr>
<tr>
<td>fendiline</td>
<td>triflupromazine</td>
<td>0.83</td>
</tr>
<tr>
<td>nelfinavir</td>
<td>abemaciclib</td>
<td>0.83</td>
</tr>
<tr>
<td>revaprazan</td>
<td>thioridazine</td>
<td>0.83</td>
</tr>
<tr>
<td>mefloquine</td>
<td>triflupromazine</td>
<td>0.83</td>
</tr>
<tr>
<td>nicardipine</td>
<td>raloxifene</td>
<td>0.83</td>
</tr>
<tr>
<td>abemaciclib</td>
<td>olmutinib</td>
<td>0.83</td>
</tr>
</tbody>
</table>
4.2 Biological Validation

From the top twenty synergistic predictions, I was only able to obtain reagents to test 18. Combination assays were performed as described in Section 3.4. After filtering for combinations with $\beta < 1$ and a maximum of 20 counts of cell survival values equal to 0 or 100, only two synergistic combinations were found. The $\beta$ values for these combinations are shown in Table 4.2. Results from the sixteen remaining combinations (though excluded from this discussion) are in the supporting information.

Focusing on Figure 4.3 we observe that neither melitracen nor mebhydrolin are particularly able to rescue SARS-CoV-2 infected cells by themselves. Indeed, melitracen and mebhydrolin provide a maximum of 12.9% and 16.4% protection when acting independently. However, we notice that at 0.12 $\mu$M of melitracen and 0.37-3.3 $\mu$M for mebhydrolin, the cells show an unexpected increase in cell survival. We confirm this increase as a synergistic effect by looking at the $\Delta$ Bliss matrix in Figure 4.4.
When melitracen and fingolimod are tested as in Figure 4.5, neither agent is able to independently rescue a majority of infected cells—melitracen providing a maximum cell survival of 32.9% and fingolimod of 12.1%. However, in combination, these agents show one row and one column where cell survival exceeds 50%. This occurs at the column where fingolimod is 3.3 μM and at the row where melitracen is 1.1 μM. Again, we can confirm this suspected region of synergy by inspecting the corresponding Δ Bliss matrix (Figure 4.4).

Figure 4.2: External Dataset Classification Metrics.

![Image of bar chart showing external data performance metrics with error bars for accuracy, recall, precision, and F1 score.](image-url)
Figure 4.3: Combination 4 Percent Cell Survival.

Figure 4.4: Combination 4 ΔBliss.
Figure 4.5: Combination 7 Percent Cell Survival.

Figure 4.6: Combination 7 ΔBliss.
Table 4.2: Experimental $\beta$.

<table>
<thead>
<tr>
<th>Drug X</th>
<th>Drug Y</th>
<th>Experimental $\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mebhydrolin</td>
<td>melitracen</td>
<td>0.93</td>
</tr>
<tr>
<td>melitracen</td>
<td>fingolimod</td>
<td>0.92</td>
</tr>
</tbody>
</table>

While I explore the reasons why these agents might exhibit synergy in Section 4.3, I address the discrepancy between the ML models and the assay results in the remainder of this section. In particular, I focus on three reasons:

1. **Dataset Size**

   In total, 176 combinations were used to develop and test my models (132 for training and 44 for testing). I suspect this number is far less than what is required to obtain high quality predictions for the problem at hand. In fact, for some applications of ML 1000 samples per class are recommended. That would suggest that I would require 3000 combinations (1000 for each of antagonistic, synergistic, and additive categories) to meet this standard.

2. **Unbalanced Dataset**

   The proportion of labels in our dataset was highly imbalanced. Out of 176 combinations, 40 were synergistic, 128 were additive, and 8 were antagonistic. Based on this fact, my model may tend to over-classify samples as additive. In terms of $\beta$, my models will likely predict values which are in the neighborhood of 1. In order to get better model performance, more highly synergistic and antagonistic combinations are required.

3. **Molecular Descriptors**

   As described in Section 2.1, 0D, 1D, and 2D descriptors are often characterized by high degeneracy—that is, even dissimilar molecules might have similar features. My
models operate under the assumption that the chosen features can sufficiently differentiate molecules in the context of SARS-CoV-2 infection. However, it is possible that information about the 3D structure of compounds and even information about targets could be incorporated to improve model performance. In particular, I want to focus on the range of targets which may be acted upon. Looking at table 1.1, there are upwards of 20 SARS-CoV-2 proteins which may be targeted. For my future work, I intend to harness the potential of these varied targets to discover if new medications with various mechanisms of action may be uncovered. If we consider the antivirals which have been approved in the United States (Section 1.2.2), then we can appreciate that there is room for medications which combat COVID-19 through many different avenues.

4.3 Mechanisms of Action

To understand why the two combinations in Table 4.2 might be synergistic, it is worth exploring the known mechanisms of action of Mebhydrolin and Fingolimod. Using these as a launching point, I hope to highlight the need to study Melitracen and its unknown mechanism of action in more depth as it relates to SARS-COV-2.

4.3.1 Mebhydrolin

Mebhydrolin is a antihistamine used to treat allergies. In terms of COVID-19, mebhydrolin may fight against viral infection by acting as a Functional Inhibitor of Acid Sphingomyelinase (FIASMA). [142, 143, 144] The work done by Carpinteiro et al. suggests that S binding to ACE2 activates Acid Sphingomyelinase. In turn, Acid Sphingomyelinase activity converts Sphingomyelinase on the host cell membrane into Ceramide which serves to cluster ACE2 receptors. Functionally, this clustering is thought to facilitate viral entry into host cells; therefore, inhibitors of Acid Sphingomyelinase are thought to make viral entry into host cells more difficult.
### 4.3.2 Fingolimod

Fingolimod is a medication used to treat multiple sclerosis. From NCATS data, fingolimod was shown to inhibit both entry of SARS-CoV-2 into cells as well as the proteolytic processing of pp1a and pp1ab. Specifically, fingolimod has been shown to reduce SARS-CoV-2 Pseudotyped particle entry. A SARS-CoV-2 pseudotyped particle is an agent that resembles SARS-CoV-2 but does not replicate in host cells. Also, fingolimod has been shown to reduce 3CL (nsp5) enzymatic activity necessary for the cleavage of pp1a and pp1ab into individual nsps. Together, these two activities explain how fingolimod increases cell survival in SARS-CoV-2 infected cells.
4.3.3 Melitracen

Melitracen is a tricyclic antidepressant (TCA) with a mechanism of action against SARS-CoV-2 that has not been yet elucidated. Even so, data from NCATS shows that melitracen can inhibit SARS-CoV-2 pseudotyped particle entry. From my work, I can hypothesize that melitracen does not act in the same manner as mebhydrolin or fingolimod because melitracen was shown to be synergistic when used with these agents. With this knowledge, it is worth investigating exactly how melitracen can increase cell survival in SARS-CoV-2 infected cells.

![Figure 4.9: Melitracen Chemical Structure.](image)
Chapter 5

Conclusion and Future Work

5.1 Conclusion

In summary, my work uses NCATS data on previously tested COVID-19 drug combinations to train and test two ML models. The top-scoring MLP model was also evaluated using an external dataset. Based on satisfactory MLP performance, I assumed my model would exhibit comparable performance on a dataset of repurposed drugs. However, out of 18 combinations predicted by my MLP model, only two showed synergistic activity. The specific mechanism of action of one of these agents, melitracen, has not been uncovered and might be a worthwhile topic to investigate. For my future work, I hope to continue exploring novel therapies against COVID-19. In particular, my work will make use of knowledge graphs to better represent molecules and their activities. I elaborate more about specific aspects of knowledge graphs that may aid my research in the following sections.

5.2 Specific Aims

1. I plan to compile information about the specific mechanisms of action of drugs. This entails populating my knowledge graph with information about drugs, viral proteins, and host proteins.

2. I plan to utilize 3D descriptors in the future. This means I need a strategy to generate suitable conformers—a difficult problem, in and of itself in terms of computation and/or storage.
3. I aim to use graph-based machine learning algorithms to uncover new therapies against SARS-CoV-2. These algorithms take advantage of the fact that the input structure of data is already in graph form.

5.3 Potential Solution

I intend to integrate diverse sources of information which contribute in a non-redundant way to the problem of uncovering novel medications for SARS-CoV-2. One strategy to accomplish this is to compile information in a knowledge graph. In such a representation, I envision individual molecules and targets as nodes of the graph and interactions as edges.

5.3.0.1 Compiling Mechanistic Information

There exists many sources of biochemical information—some similar and some disparate. I hope to include the following data sources taking care to combine similar information when necessary:

1. NCATS
2. PubChem
3. ChEMBL
4. Uniprot

5.3.0.2 Generating 3D descriptors

Drugs are dynamic structures that may adopt many different conformers depending on their chemical environment. For drug-discovery applications, the process of conformer generation is important because understanding the three dimensional arrangement of atoms gives insight into specific drug-target interactions. Sometimes 3D structure information is known for a drug in the context of a particular target. Sometimes, this information is not known. I
hope to assemble my knowledge graph in a way that incorporates experimentally validated conformers when applicable and chooses adequate conformer generation algorithms when necessary. In this way, I could leverage the power of 3D descriptors.

5.3.0.3 Using Graph-Based Machine Learning on PyG

Graph-based machine learning (otherwise known as Geometric Deep Learning) is a subfield of ML in which the inherent structure of input data is used to learn. Because various problems in biochemistry depend on the relationships between various actors (think about the interactions between all of the SARS-CoV-2 proteins in Table 1.1) it is often helpful to think of a problem in terms of these relationships. PyG is a Python library which can use graph representations to uncover new information about the dataset. It is my suspicion that with PyG, I will be able to find novel drugs which bind unexpectedly to SARS-CoV-2 proteins or perhaps discover synergistic combinations of drugs which have not been previously tested.

5.4 Time Schedule
Table 5.1: Tentative Timeline for Future Work.

<table>
<thead>
<tr>
<th>Dates</th>
<th>Tasks to Complete</th>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2022 - Aug. 2022</td>
<td>Compile and clean information about drug targets and molecules</td>
<td>Ensure information is stored and organized logically on lab resources so team members can continue work if necessary</td>
</tr>
<tr>
<td>Aug. 2022 - Nov. 2022</td>
<td>Develop knowledge graph using Neo4j and use result for graph based machine learning</td>
<td>Every drug should be associated with pre-calculated features and every target should have a structure if possible</td>
</tr>
<tr>
<td>Nov. 2022 - Dec. 2022</td>
<td>Standardize Workflow on Knime</td>
<td>My workflow should be incorporated onto Knime so new lab members do not need to be coding experts to contribute</td>
</tr>
<tr>
<td>Jan. 2023 - March 2023</td>
<td>Testing <em>in vitro</em></td>
<td>Order chemicals and coordinate with partner labs to obtain results in a timely manner</td>
</tr>
<tr>
<td>March 2023 - Oct. 2023</td>
<td>Work on dissertation</td>
<td>Produce a near-complete dissertation draft</td>
</tr>
<tr>
<td>Nov. 2023</td>
<td>Defend dissertation</td>
<td>Dissertation defense</td>
</tr>
<tr>
<td>Dec. 2023</td>
<td>Complete final version of dissertation</td>
<td>Submit final dissertation</td>
</tr>
</tbody>
</table>
Chapter 6

Supplemental Information

6.1 RDkit Descriptors List

['MaxEStateIndex', 'MinEStateIndex', 'MaxAbsEStateIndex',
'MinAbsEStateIndex', 'qed', 'MolWt', 'HeavyAtomMolWt', 'ExactMolWt',
'NumValenceElectrons', 'NumRadicalElectrons', 'MaxPartialCharge',
'MinPartialCharge', 'MaxAbsPartialCharge', 'MinAbsPartialCharge',
'FpDensityMorgan1', 'FpDensityMorgan2', 'FpDensityMorgan3', 'BCUT2D_MWHI',
'BCUT2D_MWLOW', 'BCUT2D_CHGHI', 'BCUT2D_CHGLO', 'BCUT2D_LOGPHI',
'BCUT2D_LOGPLOW', 'BCUT2D_MRHI', 'BCUT2D_MRLOW', 'BalabanJ', 'BertzCT',
'Chi0', 'Chi0n', 'Chi0v', 'Chi1', 'Chi1n', 'Chi1v', 'Chi2n', 'Chi2v',
'Chi3n', 'Chi3v', 'Chi4n', 'Chi4v', 'HallKierAlpha', 'Ipc', 'Kappa1',
'Kappa2', 'Kappa3', 'LabuteASA', 'PEOE_VSA1', 'PEOE_VSA10', 'PEOE_VSA11',
'PEOE_VSA12', 'PEOE_VSA13', 'PEOE_VSA14', 'PEOE_VSA2', 'PEOE_VSA3',
'PEOE_VSA4', 'PEOE_VSA5', 'PEOE_VSA6', 'PEOE_VSA7', 'PEOE_VSA8',
'PEOE_VSA9', 'SMR_VSA1', 'SMR_VSA10', 'SMR_VSA2', 'SMR_VSA3', 'SMR_VSA4',
'SMR_VSA5', 'SMR_VSA6', 'SMR_VSA7', 'SMR_VSA8', 'SMR_VSA9', 'SlogP_VSA1',
'SlogP_VSA10', 'SlogP_VSA11', 'SlogP_VSA12', 'SlogP_VSA2', 'SlogP_VSA3',
'SlogP_VSA4', 'SlogP_VSA5', 'SlogP_VSA6', 'SlogP_VSA7', 'SlogP_VSA8',
'SlogP_VSA9', 'TPSA', 'EState_VSA1', 'EState_VSA10', 'EState_VSA11',
'EState_VSA2', 'EState_VSA3', 'EState_VSA4', 'EState_VSA5', 'EState_VSA6',
'EState_VSA7', 'EState_VSA8', 'EState_VSA9', 'VSA_EState1', 'VSA_EState10',
...
6.2 RDkit Descriptor Families

**BCUT2D**: BCUT2D descriptors are hybrid descriptors determined by forming an adjacency matrix (a tabular representation of the connectivity of a molecule). Let us consider...
The diagonal elements are atomic contributions for a selected property—in this case molecular weight. The off-diagonal entries represent the reciprocal of the square root of the bond order between atoms. Having formed the adjacency matrix, the eigenvalues are calculated. In the case of BCUT2D_MWHI, the highest eigenvalue is reported. [145]

**BalabanJ:** BalabanJ, also known simply as J, gives a measure of a molecule’s branching. More branched molecules have higher J values. [146]

**BertzCT:** BertzCT is a topological index which quantifies the “complexity” of a molecule. This index is the sum of measures for the complexity of bonding and the distribution of heteroatoms in a molecule. [147]

**Chi:** Chi Connectivity Indices are measures related to the paths formed by the heavy atoms in a molecule. RDKit has three types of Chi Connectivity Indices—simple, valence and nVal. Each of these is associated with different path lengths. For example, Chi0 is a simple Chi index with a path length of zero while Chi1n is a nVal type index with a path length of one. For more information on the Chi Connectivity indices, the interested reader is encouraged to consult the work done by Kier and Hall. [148]

**EState_VSA:** EState_VSAs are hybrid descriptors which aggregate van der Waals surface areas (VSAs) for individual atoms of a molecule that fall under a particular Electrotopological State (EState) bin. For example, if an atom has an Estate < -0.39, the VSA of the atom is added to the EState_VSA1 bin. Information about the ranges for each bin can be found in the RDKit documentation.

**ExactMolWt:** Molecular weight is the sum of individual atomic weights for atoms in a molecule.

**FpDensityMorgan:** The family of descriptors known as FpDensityMorgan includes those that calculate the Morgan Fingerprint of a molecule and divide the count of non-zero entries by the heavy molecular weight of a molecule.

**FractionCSP3:** FractionCSP3 is the percentage of carbon atoms that are sp³ hybridized.
**HallKierAlpha:** Hall-Kier Alpha is a measure of the shape of a molecule which incorporates information about the number of bonds between heavy atoms, the heavy atom count, and the radii of heavy atoms. [148]

**HeavyAtomCount:** The heavy atom count of a molecule is the number of non-hydrogen atoms.

**HeavyAtomMolWt:** Heavy atom molecular weight is the sum of molecular weights for non-hydrogen atoms.

**Ipc:** The information content for polynomial coefficients (Ipc) is a descriptor which uses an adjacency matrix to capture information about the structure of a molecule. Low Ipc values are associated with high branching. [149]

**LabuteASA:** Labute’s Approximate Surface Area (ASA) is a method used to calculate the surface area of a molecule. [150]

**EStateIndex:** EState is an index which combines the electronic character and topological environment of a particular atom. [151] The maximum, minimum, absolute value maximum and absolute value minimum EStates of atoms in a molecule are reported.

**PartialCharge:** Gasteiger partial charges are calculated for individual atoms in a molecule. The maximum, minimum, absolute value maximum and absolute value minimum Gasteiger charges are reported. [152]

**MolWt:** Molecular weight is the sum of individual atomic weights for atoms in a molecule.

**NHOHCount:** This descriptor is the count of NH residues or OH residues in a molecule.

**NOCount:** This descriptor is the count of nitrogen atoms or oxygen atoms in a molecule.

**NumAliphaticCarbocycles:** This descriptor is the count of aliphatic carbocycles in a molecule.

**NumAliphaticHeterocycles:** This descriptor is the count of aliphatic heterocycles in a molecule.
**NumAliphaticRings:** This descriptor is the count of the number of aliphatic rings in a molecule.

**NumAromaticCarbocycles:** This descriptor is the count of the number of aromatic carbocycles in a molecule.

**NumAromaticHeterocycles:** This descriptor is the count of the number of aromatic heterocycles in a molecule.

**NumAromaticRings:** This descriptor is the count of the number of aromatic rings in a molecule.

**NumHAcceptors:** This descriptor is the count of the number of hydrogen bond acceptors in a molecule.

**NumHDonors:** This descriptor is the count of the number of hydrogen bond donors in a molecule.

**NumHeteroatoms:** This descriptor is the count of the number of heteroatoms in a molecule.

**NumRotatableBonds:** This descriptor is the count of the number of rotatable bonds in a molecule.

**NumSaturatedCarbocycles:** This descriptor is the count of the number of saturated carbocycles in a molecule.

**NumSaturatedHeterocycles:** This descriptor is the count of the number of saturated heterocycles in a molecule.

**NumSaturatedRings:** This descriptor is the count of the number of saturated rings in a molecule.

**RingCount:** This descriptor is the count of the number of rings in a molecule.

**MolLogP:** The partition coefficient P, is a measure of the lipophilicity of a molecule. More lipophilic molecules have higher (more positive) logP values. The Wildman-Crippen LogP value (used by RDKit) takes into account individual atomic contributions to logP. [153]

**MolMR:** Molar refractivity (MR) is a measure of the polarizability of one mole of
matter. The Wildman-Crippen MR value (used by RDKit) takes into account individual atomic contributions to MR. [153]

**PEOE_VSA:** PEOE_VSAs are hybrid descriptors which aggregate VSAs for individual atoms of a molecule that fall under a particular Partial Equalization Of Orbital Electronegativity (PEOE) bin. For an overview of how to determine PEOE, one can consult the work done by Gasteiger and Marsili. [152]

**SMR_VSA:** SMR_VSAs are hybrid descriptors which aggregate VSAs for individual atoms of a molecule that fall under a particular MR bin. The method used to determine MR is the one discussed by Wildman and Crippen. [153]

**SlogP_VSA:** SlogP_VSAs are hybrid descriptors which aggregate VSAs for individual atoms of a molecule that fall under a particular logP bin. The method used to determine logP, is the one discussed by Wildman and Crippen. [153]

**TPSA:** Topological Polar Surface Area (TPSA) is an estimate of the polar surface area of a molecule calculated using chemical group contributions as described in the work done by Ertl and colleagues. [154].

**VSA_EState:** VSA_EStates are hybrid descriptors which aggregate EStates for individual atoms of a molecule that fall under a particular VSA bin. The method used to determine EState, is the one discussed by Kier and Hall. [151]

**Fragments:** Fragments are descriptors that count the number of predefined residues in a molecule. For example, fr_Al_COO represents the number of aliphatic carboxylic acids in a molecule. Consult RDKit documentation for more information.

**qed:** The descriptor known as quantitative estimation of drug-likeness (qed) combines several different descriptors into one metric that predicts the drug-like behavior of molecules. Molecules that are drug-like have qed values close to 1 and those that are not drug-like have qed values close to 0. [155]

### 6.3 Non-Synergistic Drug Combinations
References


Xin Zou et al. “Single-cell RNA-seq data analysis on the receptor ACE2 expression reveals the potential risk of different human organs vulnerable to 2019-nCoV in-


[87] Xuping Xie et al. “Neutralization of SARS-CoV-2 spike 69/70 deletion, E484K and N501Y variants by BNT162b2 vaccine-elicited sera”. In: Nature medicine 27.4


[104] Calvin J. Gordon et al. “Remdesivir is a direct-acting antiviral that inhibits RNA-dependent RNA polymerase from severe acute respiratory syndrome coronavirus 2 with high potency”. In: *The Journal of Biological Chemistry* 295 (20 May 2020), p. 6785. ISSN: 1083351X. DOI: 10.1074/JBC.RA120.013679. URL: /pmc/articles/


A. R. Finkelmann, A. H. Göller, and G. Schneider. “Robust molecular representations for modelling and design derived from atomic partial charges”. In: Chemical Communications 52.4 (2015), pp. 681–684. ISSN: 1364548X. DOI: 10.1039/C5CC07887C. URL: https://pubs.rsc.org/en/content/articlelanding/2016/cc/c5cc07887c


Curriculum Vita

Jason Sánchez graduated from Rice University in 2018 with a Bachelor of Science in Biochemistry & Cell Biology. He joined The University of Texas at El Paso (UTEP) in 2019 and is anticipated to receive his masters degree in Computational Sciences (CPS) in May 2022 before continuing onto his PhD. Jason worked as a teaching assistant at UTEP until 2022 when he became a research assistant. Jason published in the *Journal of Chemical Information and Modeling* in 2021. His work was geared toward using machine learning to decipher the chemical features which lead to G Protein bias. Jason’s current work focuses on COVID-19, but his research interests are broadly in data science, high performance computing, cheminformatics, and drug discovery.