Deep Learning Models for Variant Pathogenicity Prediction

Deep Learning Models for Variant Pathogenicity Prediction

A DISSERTATION PRESENTED
BY
Ralph R. Estanbouileh
TO
The Departments of Chemical and Physical Biology and Computer Science

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Bachelor of Arts and Sciences
IN THE SUBJECTS OF
Chemical and Physical Biology and Computer Science

Harvard University
Cambridge, Massachusetts
March 2021

Thesis advisor: Professor Arjun K. Manrai Ralph R. Estanboulieh

Deep Learning Models for Variant Pathogenicity Prediction

Abstract

Technological advancements in DNA sequencing have made it a mainstay in the clinic in the form of targeted genetic testing as well as whole exome and whole genome sequencing (WES and WGS, respectively). With increased use comes a growing need to interpret the abundance of data being generated and the myriad variants being discovered. Many computational methods have been developed to address this issue, with varying levels of success. My goal in this thesis is to build on such methods by altering the underlying the models, the learning algorithms, and the data being used, and then apply them to the task of clinical-grade variant pathogenicity classification. To do so I first review and compare the methods that have been developed so far, trying to identify a common pattern of strengths, weaknesses, and aspects to account for. Then, I reproduce a foundational method developed for the interpretation of hypertrophic cardiomyopathy-related disease, PolyPhen-HCM. Finally, using the insights learned from both the comprehensive review and the redesigning and in-depth analysis of PolyPhen-HCM, I introduce deep learning models that address, through their improved architectures and data, some of the most salient issues that methods in variant interpretation have to deal with.

Contents

Listing of figures

To my families at home in Syria, and across the Atlantic.

Acknowledgments

I am extremely thankful to all of those who have supported me along the way, including my thesis advisor, Prof. Raj Manrai , and my concentration advisor, Dr. Elena Rivas. I am also incredibly thankful for all those who have mentored me and taught me so much over the past many years, including Prof. Andrew Murray, Prof. Sean Eddy, Dr. Domninic Mao, Prof. Rachelle Gaudet, Prof. Douglas Melton, Dr. Sharon Nadav, and many, many others. I am also deeply thankful to my family for all the support they have provided me, despite the distance that has separated us for so long.

1

Introduction

The low costs and improved streamlining of whole genome and exome sequencing are making these methods increasingly accessible and instrumental tools for the diagnosis and treatment of Mendelian disorders in the clinic. The growth of the National Center for Biotechnology Information (NCBI) Genetic Testing Registry is a testament to how widespread genetic testing has become in clinical decision-making. As of this writing, it contains 76,273 genetic tests for 16,396 conditions and 18,695 genes – and counting.

This remarkable increase in access to sequencing tools has deluged the field of clinical genomics with tens of millions of newly discovered variants over the past few years, bringing with it a growing need to interpret and understand this abundant data. Scalable and accurate tools are urgently needed to improve the way we diagnose and treat inherited disorders, elucidate new underlying genetic mechanisms, and synthetically engineer proteins

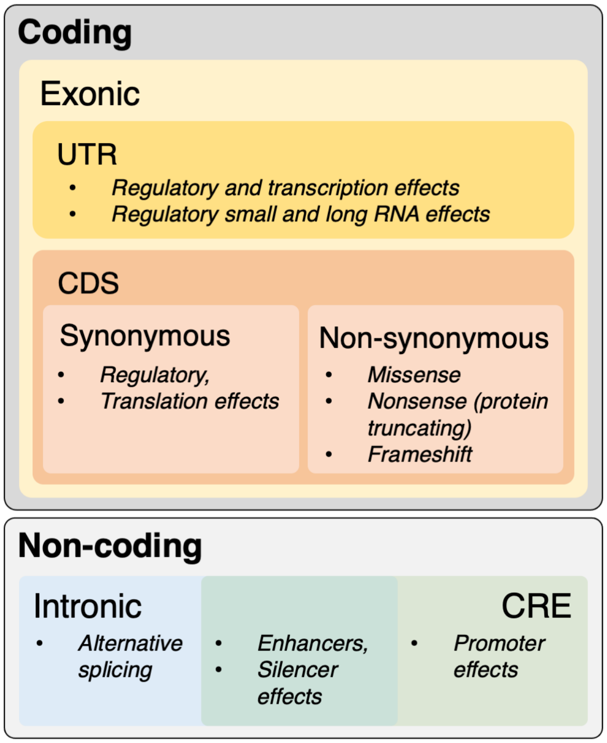


FIGURE 1.1: **Types of variants and their effects, oversimplified.** This diagram by no means lays out the entire picture, but it is detailed enough for our purposes. UTR: Untranslatd Region; CDS: Coding Sequence; CRE: *Cis*-regulatory element.

with new functions.

1.1 WHAT IS VARIANT INTERPRETATION?

Before we delve into variant interpretation, we must clarify the meaning of a variant. Before the term "variant" was adopted in genetics, "mutation" was used to describe a permanent change in the nucleotide sequence of a genome, and "polymorphism" was defined as a benign permanent change seen in at least 1% of the population. To avoid any confusion regarding assumptions about the pathogenicity (or lack thereof) of a sequence change, the neutral term "variant" was recommended to replace both "mutation" and "polymorphism" (Richards et al., 2015). In any case, the sequence change is relative to the most common, or reference, nucleotide at a particular position in the population. (But this is not a satisfying reference: what is considered a population is still a topic of debate.)

There are many types of variants, depending on the sequence change, its location, and its consequences. If a variant affects a single nucleotide, it is called a Single Nucleotide Variant (SNV), or Single Nucleotide Polymorphism (SNP) if it found in at least 1% of the population. If a variant affects multiple nucleotides at the same time, changing all of them without changing the length of the sequence, it is called a Multiple Nucleotide Polymorphism (MNP). If an MNP also changes the length of the sequence, it is called an insertion-deletion (indel). If a variant affects entire chromosomal loci (e.g. inversions, translocations, duplications, and more), it is called a structural variant. Although structural variants are key features of many Mendelian disorders, the approaches this thesis reviews and proposes do not account for them. Figure 1.1 outlines some of the most common variant effects according to where they occur.

In this thesis, I use the terms "allele" and "variant" interchangeably to describe most sequence alterations, while occasionally using "mutation" and "polymorphism" to describe variants that are known to be pathogenic or benign, respectively. Because most of the interpretation methods deal with SNVs and short indels, it is reasonable to assume that a variant instance is an SNV or an short indel.

Interpreting a variant generally means estimating the probability that it is causal of disease, this probability is also known as the *pathogenicity* of a variant. Currently, pathogenicity estimation is qualitative. According to the recent recommendations of the American College of Medical Genetics and Genomics (ACMG) and the Association for Molecular Pathology (AMP), a variant can either be 'likely benign', 'benign', 'likely pathogenic', 'pathogenic', or a 'variant of unknown significance' (VUS) (Richards et al., 2015). Just over 40% of the variants in the NCBI ClinVar database are classified as VUS, making this category the largest one in the set — not including the several millions of VUSs discovered by recent exome and genome aggregation consortia (Lek et al., 2016; Karczewski et al., 2020).

Variant prioritization is another side of the same coin wherein genetic variants are ranked according to how causal they are to the development of disease (Eilbeck et al., 2017). It is important to distinguish between variant prioritization and gene prioritization; both aim to elucidate the causes for an inherited disease or predisposition but the former is specific to variants (which might span many different genes), whereas the latter is specific to genes (which might contain many variants).

Another important distinction to make is that between *pathogenicity* and *penetrance*. Pathogenicity describes how causal a variant is to a particular disease phenotype, whereas penetrance is the probability that an individual will display that disease phenotype *given* that they have the variant. The difference between pathogenicity and penetrance can be likened to that between causality and correlation: certainly, a variant might correlate highly with a disease (thereby increasing its penetrance), but because correlation does not imply causality, a variant's high penetrance does not necessarily imply that it is pathogenic (e.g. disease-causing).The reverse is worth emphasizing too – pathogenicity does not imply high penetrance. Many purportedly pathogenic variants do not have high

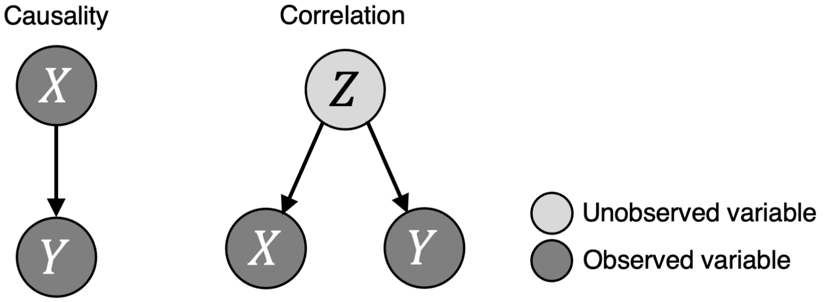


Figure 1.2: **Correlation does not imply causation.** Two simple examples in plate notation. On the left, we say that X causes Y. On the right, we say that X and Y are correlated and that neither causes the other because there is an unobserved variable Z that is causal to both. Given Z, X and Y are independent of each other, and they are only correlated when unconditioned on Z.

penetrance. Indeed pathogenicity can be a very elusive and ambiguous concept.

Ideally, we would like to design a computational method that takes a variant for input and returns as output the probability that this variant is pathogenic. Alternatively, this model could take an entire sequence and return the same type of output. Throughout this thesis, we will show that one can do even better.

1.2 Why Variant Interpretation is Hard

Pathogenicity estimation can be formalized using the language of causal inference. This is inherently difficult because of what is known as the fundamental problem of causal inference: we only get to experience one universe. Establishing causality is done by exposing a system to multiple types of treatments and observing the average outcomes, based on which we can claim that treatment X is causal to outcome Y. However, this is impossible as long as the theory of parallel universes is but a fantasy: there is no way to simultaneously expose *and* not expose a system to a treatment. A common workaround is to run double-blind randomized control trials (RCTs) in which participants are randomly assigned to either the placebo group or the treatment group, and then compare rates of success (however defined) while making strong assumptions about the independence — or noninterference — of outcomes. This independence assumption breaks down in many cases where the outcome of a particular participant depends on that of another one, such as in a classroom setting with dense student interactions or a vaccine study where participants given the vaccine indirectly protect those who are only given the placebo. In cases such as studying the effects of a mutation in an individual, RCTs would raise a host of ethical issues, and so observational studies are the only option to study disease-causing variants.

The difference between causation and correlation becomes clearer when looking at a simple graphical model (also known as a directed acyclic graph, or DAG), which is a visual representation of the causal story that generates the data (see Figure 1.2). An arrow going from X to Y means that X is sampled first, and that Y is sampled in a manner dependent on the value of X. Loosely speaking, we say that X *causes* Y if there is some path from X to Y. If X and Y are not linked this way, they may correlate, but that is still not causality. The same way, the correlation between a genetic variant and a disease does not imply that this variant causes that disease.

This inevitable reliance on causal DAGs is a source of great difficulty, and the factors at play are manifold. First comes the size of the search space. As it turns out, the number of possible DAG arrangements increases *super-exponentially* with the number of variables in the model. With only 6 nodes, there are almost 3.8 million possible DAG configurations, ignoring the potential additional hidden (unobserved) variables. The human population harbors at least hundreds of thousands of non-synonymous SNPs (nsSNPs), and each person is thought to hold at least 24,000 nsSNPs (Cargill et al., 1999). Two very large exome and genome aggregation projects, namely the Exome Aggregation Consortium (ExAC) and the Genome Aggregation Database (gnomAD), demonstrate an even greater degree of variation. ExAC detected 7.4 million high-quality variants through the whole exome sequencing of over 60,000 humans, while gnomAD found around 280 million variants through the whole exome *and* genome sequencing of over 141,000 individuals (Lek et al., 2016; Karczewski et al., 2020). In the ExAC data set, 72% of the unique variants had never been seen before, as they were absent from both the 1000 Genomes and the ESP data sets. With this number of variables, the number of possible DAG configurations is probably much greater than the number particles in the universe, making the search space intractably large. And because most of this variation is thought to be benign (having survived purifying selection), finding the variants that are truly causal to disease is like finding a hay stalk in a stack of needles.

One reasonable idea to solve this issue is to start with, and only consider, DAGs that approximate biologically relevant interactions — or the interactome. The interactome is the map of all interactions amongst all chemical and physical factors related to life (e.g. proteins, nucleic acids, small molecules, inorganic compounds, temperature, pressure, light, and many more). The major challenge with this approach is that our understanding of the interactome is incomplete. It is estimated that the human protein-protein interaction network alone is composed of about 650,000 pairwise interactions and that we have only identified about 10% of them (Hart et al., 2006; Stumpf et al., 2008). Without a blueprint of the interactome, it becomes very hard to model causal networks amongst genes, proteins, and the myriad variants that underlie many polygenic diseases that affect a large proportion of the human population, such as diabetes, hypertension, and cardiomyopathy. In fact, we know the genetic mechanisms for only

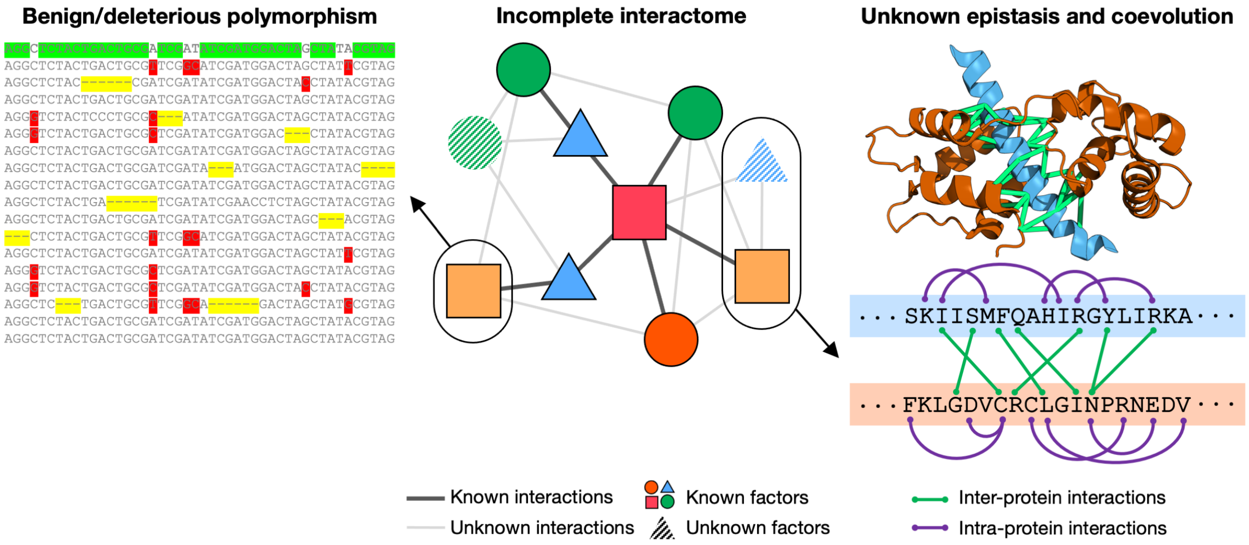

Benign/deleterious polymorphism
AGGCTCTACTGACTGCGATCGATATCGATGGACTAGCTATACGTAG
AGGCTCTACTGACTGCGTTCGGCATCGATGGACTAGCTATTCGTAG
AGGCTCTAC-------CGATCGATATCGATGGACTACCTATACGTAG
AGGCTCTACTGACTGCGATCGATATCGATGGACTAGCTATACGTAG
AGGGTCTACTCCCTGCGC---ATATCGATGGACTAGCTATACGTAG
AGGGTCTACTGACTGCGCTCGATATCGATGGAC---CTATACGTAG
AGGCTCTACTGACTGCGATCGATATCGATGGACTAGCTATACGTAG
AGGCTCTACTGACTGCGATCGATA---ATGGACTAGCTATAC----
AGGCTCTACTGACTGCGATCGATATCGATGGACTAGCTATACGTAG
AGGCTCTACTGA------TCGATATCGAACCTCTAGCTATACGTAG
AGGCTCTACTGACTGCGATCGATATCGATGGACTAGC---ACGTAG
---CTCTACTGACTGCGTTCGGCATCGATGGACTAGCTATACGTAG
AGGCTCTACTGACTGCGATCGATATCGATGGACTAGCTATTCGTAG
AGGGTCTACTGACTGCGCTCGATATCGATGGACTAGCTATACGTAG
AGGGTCTACTGACTGCGCTCGATATCGATGGACTACCTATACGTAG
AGGCTC---TGACTGCGTTCGGCA------GACTAGCTATGCGTAG
AGGCTCTACTGACTGCGATCGATATCGATGGACTAGCTATACGTAG
AGGCTCTACTGACTGCGATCGATATCGATGGACTAGCTATACGTAG
Incomplete interactome
Unknown epistasis and coevolution
· · · SKIISMFQAHIRGYLIRKA · · ·
· · · FKLGDVCRCLGINPRNEDV · · ·
Known interactions
Unknown interactions
Known factors
Unknown factors
Inter-protein interactions
Intra-protein interactions

FIGURE 1.3: **Why variant interpretation is hard, visualized.** The center panel depicts a sub-graph of the incomplete interactome, where most edges (interactions) and a few factors remain undiscovered. The leftmost panel shows a multiple sequence alignment (MSA) for one of the factors in the network, the MSA shows substantial polymorphism — some of which might be deleterious. The rightmost panel shows an example of coevolving sites within and across proteins that interact with each other. The protein fold shown here is the scallop myosin in the near-rigor conformation (PDB 1KK7); the coil in blue is a section of the tail of the myosin heavy chain, and the orange structure is part of the myosin light chain.

about half of all known Mendelian disorders (Chong et al., 2015), and there are over 3000 hereditary conditions whose genetic basis has yet to be elucidated.

Modeling the consequences of variants is rarely straightforward. Depending on a variant's position in the genome, many consequences are possible. Besides being capable of modifying RNA splicing sites, intronic mutations can have significant effects on epistasis and epigenetics, which can alter the tissue-specific expression levels of a gene and its isoforms. Exonic variants can lead to similar effects if they occur in regulatory elements, and even more once the gene transcript is translated. Coding mutations are *synonymous* when they affect the gene of a protein without affecting the protein's primary structure (when the mutated codon encodes the same amino acid as the wild-type codon), and they are *non-synonymous* if the protein's primary structure is altered. A synonymous variant might have effects on regulation of transcription and translation of an mRNA, while a non-synonymous coding variant can either cause protein loss-of-function, gain-of-function, or no change, in addition to potential effects on regulation. Intergenic variants may affect the expression levels of many genes if they occur in the *cis*-regulatory elements specific for these genes. Many non-coding and non-regulatory regions of the genome have unknown functions, making it even harder to predict the effects of variants occurring there. In addition, variants that affect the genes for small non-coding RNA (snRNA) or long intergenic non-coding RNA (lincRNA) can also have very ambiguous consequences.

The consequences of many variants are also not necessarily additive. Proteins that interact with each other and with nucleic acids tend to coevolve, therefore making their sequence positions highly interdependent. This phenomenon, known as *epistasis*, has been found to play a major role in molecular evolution (Breen et al., 2012; McCandlish et al., 2016). Examples of such coevolving sites are the interacting residues within a protein's secondary (within an α-helix or a β-sheet), tertiary (within a single protein fold), and quaternary structures (at the interfaces within a protein complex or multimer), or even the residues of a transcription factor (TF) and the nucleotide sequences of the DNA elements to which this TF binds. Understanding where coevolution happens and the constraints that it imposes on sequence space is instrumental to make accurate predictions of variant consequences. For instance, several mutations coordinated across coevolving sites can result in drastically different predictions depending on whether we know about the interdependencies between these sites. Without such knowledge, our predictions could be massively erroneous. Alas, it is hard to infer, model and encode coevolutionary interactions, and many models naïvely assume that positions within a protein or DNA sequence are independent of each other.

Moreover, the map from genotype to phenotype is not linear. In other terms, not all damaging mutations (i.e. causing proteins to lose their function or become under- or over-expressed) are pathogenic (i.e. causing disease phenotype), and not all pathogenic mutations are damaging. In the case of dominant Mendelian disorders, a damaging variant is likely to be pathogenic, but that no longer holds true for recessive and polygenic diseases in which the effects of many variants might combine and synergize to cause disease. This non-linearity adds yet another layer of complexity to variant pathogenicity prediction: a method that is unaware of it might overestimate the pathogenicity of variants that are predicted to be damaging.

Finally, environmental factors such as diet, pollution, physical exercise, and exposure to other chemical or physical factors can lead to even more variation at the phenotype level, but these considerations are not within the scope of this thesis.

To recapitulate, pathogenicity estimation is hard because causal inference is hard. The intractable causal relationships between the myriad factors, hidden variables, and unknown interactions that govern such a complicated process as pathogenicity only make this even harder. Consequently, there is no model we have for variant pathogenicity; there is no paradigm or central dogma that explains why some variants are pathogenic and why others are not. In the upcoming sections, we will focus on *what we know*; we will explore the data and the general strategies that the field has adopted to circumnavigate these challenges, as well as potential ways to improve these strategies — including redefining pathogenicity itself.

1.3 Back to the Feature: the Data

Instead of focusing on causality, all the methods we will encounter throughout this thesis are prediction-oriented; they rely on different kinds of data, heuristics, and dependencies that guide our predictions of pathogenicity. We will formalize this shift from causal thinking to prediction in the next chapter, but for now we can think of prediction as an attempt to estimate $P(D|\boldsymbol{X})$; where D is the event of having the disease and $\boldsymbol{X}$ is a vector of features for the variant in question. The purpose of this section is to give the reader a glimpse of what types of features are used by the current methods in the field. A much more detailed and rigorous review of these methods will follow in Chapter 2.

Conservation and sequence homology. The earliest methods developed for interpreting genetic variants investigate the sequences they are found in, as phylogenetic and conservation patterns usually reflect quite well whether a variant will be tolerated or not. ("Tolerated" here means benign, or not interfering with the function of a protein or a DNA sequence.) This approach taps into dense evolutionary information that has accumulated for millions of years and more. To do this, methods will generate a large multiple sequence alignment (MSA) of homologous sequences and extract information from that MSA, be it position-specific amino acid frequencies or other information with more complicated structure.

The results of conservation and homology-based analyses are highly dependent on the sequences accrued in the MSA; the optimal MSA is thought to have a large number of orthologous sequences that have identical function. Data that is not sampled independently and according to its real-world distribution will undermine the generalizability of our prediction models. A set of sequences that are too similar to each other might not reflect fully the degree of polymorphism that a gene tolerates, resulting in more 'conservative' predictions and a greater rate of false alarms. On the other hand, a set of sequences that are not similar enough to each other might completely throw off our model's predictions. Of course, it is not easy to guarantee that two sequences will have the same function without carrying out wet lab experiments, nor is it easy to ensure that sequences are sampled independently from a population. Many of the methods that rely on conservation and homology will have a sequence weighting scheme to compensate for these issues. These schemes (which we will discuss at the end of the next chapter) usually put less weight on similar sequences and more weight on unique ones.

Protein structure. With the advent of powerful and high-resolution imaging, crystallization, and magnetic resonance techniques, it has become easier to determine the structure of increasingly large and convoluted proteins. In the context of variant interpretation, protein structure data is exclusively used to predict the pathogenicity of non-synonymous coding variants. A variant sequence is usually mapped to a structure by homology search and alignment, and structural information is extracted at the variant's location, such as the crystallographic temperature factor, solvent accessibility, carbon-beta density, and even changes in Gibbs free energy (Ng and Henikoff, 2006).

Using protein structures for prediction is not without its drawbacks. Because not all proteins have known structures, not all sequences can be perfectly mapped to a protein fold, which might give inaccurate structural data about a variant residue. Protein structures are also not necessarily accurate depictions of a protein's state in a living cell. For instance, the solvent accessibility of a particular residue might change as the biochemical state of the protein changes. In addition, certain features such as the crystallographic temperature factor depend on the experimental setup (and thus comparing them across different structures might not be appropriate).

Functional genomic annotations. Other types of knowledge can be used to annotate positions in the genome, and these annotations are instrumental to interpret non-coding variation, which constitutes most of the variation observed. Databases such as GenBank, Ensembl (Aken et al., 2017), and the ENCODE consortium (Dunham et al., 2012) provide gene models that specify where genes start and end, transcribed regions, splice sites, translated and untranslated regions, promoters and transcription factor binding sites, as well as other regulatory information. Other functional annotations include domain data (i.e. transmembrane domains or ATP-binding), ChIP-seq for protein-DNA interactions, RNA-seq for transcriptional activity, DNase sensitivity for chromatin exposure, chromatin structure and histone modification, and more. However, functional annotations are not flawless either. They rely on gene models, which can vary significantly from database to database and from genome build to genome build. The start, end, or splice positions of a gene may vary between Ensembl and Genbank. Moreover, functional annotations are often the result of independent analyses of the genome which may themselves be flawed.

Population allele frequencies and population stratification. Large scale sequencing projects such as the 1000 Genomes project (McVean et al., 2012), the US National Heart, Lung and Blood Institute (NHLBI) Exome Sequencing Project, the ExAC (Lek et al., 2016), gnomAD (Karczewski et al., 2020), and more have charted the landscape of almost 10 million genetic variants from hundreds of thousands of diverse genomes and exomes. All of this data has been used to estimate variant frequencies, which can be later used as features in prediction. The frequencies that have been calculated do align with our intuition for pathogenic variation: the variants that some methods predict as deleterious tend to have very low population frequencies (Kircher et al., 2014). Another way variant frequencies are being used in databases such as ClinVar is to challenge pathogenicity claims of variants whose frequencies are larger than would be expected based on the prevalence and genetic architecture of the inherited disorder(s) they are thought to contribute to (Shah et al., 2016).

Population stratification refers to the differences in allele frequencies among different populations. Different subpopulations often exhibit different sets of common polymorphisms, and considering these differences has become increasingly important as sequencing projects target more and more diverse sets of groups. For instance, an allele with a very low average frequency might be fairly common in one subpopulation due to ancestral divergence in selective pressures.

Gene constraints. Some genes may tolerate mutations because of the inherent robustness of their sequences, while others can only deal with so much variation until they loose their function. Thus, we can say that each gene has its own constraints on potentially deleterious amino acid variation. There are several methods that try to account for such constraints. One estimates them by comparing the number of functional variants (e.g. variants that lead to changes in function or expression) with the total number of observed variants (Petrovski et al., 2013). Using the ExAC data, Lek et al. compare the observed and expected numbers of loss-of-function (LOF) variants to estimate the probability of LOF intolerance (pLI) (Lek et al., 2016).

The coarseness of a gene-wide pLI estimator takes away important regional information from within the gene. In reality, mutational constraints can differ across domains within a single gene: some domains might be more pivotal for protein function and stability than others. As a result, partitioning a gene into functionally distinct domains or subunits and calculating regional constraints for each of these domains represents a clear improvement on gene-wide constraint measures (Eilbeck et al., 2017).

1.4 What this Thesis is About: a More Complete Model

So far, we have been using a conventional definition of pathogenicity: it is the probability that a genetic variant is causal to disease. Where, as mentioned earlier in this introduction and in agreement with most of the methods that will be reviewed in the next chapter, a variant only affects a DNA sequence at one or a few (e.g. in insertion-deletion mutations) positions. We formally consider the idea of defining pathogenicity as a property of not a single variant, but a list of variants occurring in disease-relevant genomic regions. We will call such a group of variants a *variant ensemble*. The rationale for this re-definition follows.

Section 1.2 and Figure 1.3 elaborate on a few of the reasons for why variant interpretation is hard. In short, it is hard because the causal models involved are very large, dense, and mostly unknown. In particular, it is hard because of phenomena like epistasis and coevolution that couple genomic positions (or groups of them) together. Such couplings could be of any order, meaning that they do not have to be pairwise. The main argument for why pathogenicity should be a property of a variant ensemble is because of all these interactions and couplings that we leave out when we define pathogenicity otherwise.

Genes, proteins, and other biological and non-biological factors interact intensively at many levels. That is why a lot of inherited diseases and predispositions are polygenic or multifactorial, as they result from the combined action of many genes and the exposure to certain environmental factors. (Unlike monogenic diseases such as cystic fibrosis, which results from characterized mutations of the CFTR gene.) In fact, some of the most prevalent inherited diseases fall into this category: diabetes, cardiovascular diseases (including a wide array of conditions), Alzheimer's disease, and many others.

Two recent methods reviewed in this thesis consider such tangled interactions; one uses an energy-based model and direct coupling analysis to uncover pairwise coevolution trends, the other uses a generative model that accounts for sequence-wise coevolution (including pairwise and higher-order couplings). However, there is one other factor missing: the interactome. Proteins and DNA sequences that bind and interact with each other coevolve, therefore pathogenicity must also be a function of these network-based associations.

The main contributions of this thesis are twofold. First, we carry out a rigorous and comprehensive review of pathogenicity prediction methods spanning a diverse set of paradigms, while relating these methods to each other and establishing an overarching theme of approaches and a hierarchy of complexities. Second, we propose a network-based deep generative model, specifically an interactome-inspired convolutional variational autoencoder, that integrates interacting DNA and protein sequences to predict the pathogenicity of variant ensembles observed in them.

In Chapter 2, we will start by introducing important concepts in learning theory that will be useful to reason about the methods discussed later in the same chapter. This review will serve a dual purpose. The first purpose is retrospective; it is often useful to find a unifying thread or an evolving pattern of thinking about models for variant interpretation, a problem that has been poorly formulated so far. Indeed, many of the methods developed so far are devoid of a model based on more than oversimplified biology. Nevertheless, a few recent ones have started addressing this issue of model-lessness, and a review of all these approaches will teach us about how we transitioned from one approach to the other. The second purpose is prospective; once equipped with a better idea of the status quo and a first-principled model for pathogenicity prediction, we can pursue original avenues of improvement — which we will do in the last two chapters of this thesis.

2

Computational Pathogenicity Prediction: an Evolving Theme

The traditional methods by which variants are interpreted in the clinic follow *decision support frameworks* (also called *decision trees*): tools that integrate clinical context to guide medical decision-making (Doig et al., 2017). This clinical context includes the family and medical history of the patient, co-segregation with disease phenotype, presence in healthy controls, strong functional data, and more. For instance, a variant seen in multiple families of individuals affected by a disease, occurring in a gene known to play a role in the pathophysiology of said disease, and with a very low population frequency is a perfect candidate for prioritization. However, the variability and subjectivity of these frameworks and their inability to formalize causal relationships has made them less reliable than desired. For example, a recent study showed surprisingly low concordance rates among the variant interpretations provided by different genetic testing laboratories (Amendola et al., 2016). Moreover, manual methods of classification are time-consuming and do not scale up with the millions of variants to be interpreted.

Because pathogenicity is so elusive, we can only try to approximate it. The variables that are accounted for in the decision frameworks above (e.g. population frequency, family pedigree, molecular mechanisms) might simply be correlations that do not imply causation. Nevertheless, they are still useful for something else: *prediction*.

Even if knowledge about the causal network between variables implies the ability predict an outcome based on its causes, prediction does *not* require the knowledge of causal relationships. The correlations among variables still gives us some information about their joint distributions, thereby enabling us to recognize their patterns and predict the value of unobserved variables based on the values of observed ones. Even if X does not cause Y, I can still use statistical inference or machine learning algorithms to predict one from the other after having seen a lot of samples (examples) from their joint distribution $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. For example, I could do a linear regression and learn the parameters of the resulting line of best fit $\widehat{\mathbb{E}[Y|X]} = aX + b$ (the slope and the intercept).

The purpose of this chapter is to review a sample of methods that apply a diverse set of paradigms to approach pathogenicity prediction. As noted in the previous chapter, the objective of such a review is to examine the modes of thinking that underlie these methods and to find a common evolving thread that allows us to understand them better and think about — or even justify — what comes next. But before we lay out the roadmap and delve into the prediction models, it is necessary that we familiarize ourselves with some of the most important principles of prediction and machine learning. This understanding will provide a rigorous and principled context that will be greatly useful throughout the ensuing methods review. For that, we will spend the next section re-establishing some of the groundwork in computational learning theory.

2.1 A Primer in Learning Theory

There are many types of learning, the major three of which are supervised, unsupervised, and reinforcement learning. These types of learning depend on the type of data to which a model has access and how a model interacts with its environment. Briefly, supervised learning is carried out using labeled data, unsupervised learning uses unlabeled data to discover clusters and structure, and reinforcement learning tries to maximize the 'goodness' of a policy based on the rewards imposed by the environment. In this thesis, we will only worry about the first two types of learning.

Let's call the algorithm that returns the sought-after $\hat{h}$ a *learner*. This learner might be anything from linear regression to state-of-the-art deep neural networks, or anywhere in between. The learner takes for input three things:

- **The domain set:** The set of all possible inputs (also called *instances*), denoted $\mathcal{X}$. Usually, these are represented by *feature vectors* $\boldsymbol{x}$. A feature is simply an individual property of an input. In the context of variant pathogenicity prediction (henceforth "In our context"), a feature could be any information about a variant that might help us predict things about it, such as its frequency in the ExAC database (or any other genomics database), or the solvent accessibility of the resulting variant amino acid.

- **The label set:** The set of all possible labels for our data, denoted $\mathcal{Y}$. In our context, $\mathcal{Y} = \{-1, 1\}$ where 1 means benign, and -1 mean pathogenic.

- **Training data:** A finite sequence of m inputs and their known labels $\mathcal{S} = ((x_1, y_1), \ldots, (x_m, y_m))$ such that $S \subseteq \mathcal{X} \times \mathcal{Y}$. This is the data we have to tune our model's parameters. Of course, not all possible pairs are equally likely to appear in our data set; the instances we see are usually generated by some probability distribution $\mathcal{D}$ over $\mathcal{X}$. $\mathcal{D}$ depends on the generative model for the data, i.e. it depends on the distributions of the individual features and their causal relationships and correlations. The learner does not know about $\mathcal{D}$. As to the labels, we will simply assume that there is one always correct labeling function $h^* : \mathcal{X} \to \mathcal{Y}$ such that for every i, $x_i = h^*(y_i)$. This function is what the learner is trying to find. An essential assumption here is the the *i.i.d. assumption* (independent and identically distributed), which stipulates that each input-label pair in $\mathcal{S}$ is sampled independently from the other (so no correlations are seen between the data points).

After the learning process, our learner outputs a hypothesis $\hat{h} : \mathcal{X} \to \mathcal{Y}$ which maps inputs (data) to output labels (predictions). $\hat{h}$ is also called a predictor, a prediction rule, or a classifier. In our context, $\hat{h}$ represents the program that will ideally be used in the clinic to find out if a new variant is pathogenic.

What is the 'goodness' of a classifier h, and how do we measure it? The error of a classifier is defined as the probability of sampling a data point from the underlying distribution $\mathcal{D}$ over $\mathcal{X}$ that is incorrectly classified. Mathematically, the error L of a hypothesis h, given a distribution $\mathcal{D}$ and an optimal classifier h^* is

$$L_{\mathcal{D},h^*}(h) \triangleq P_{x \sim \mathcal{D}} \left(h(x) \neq h^*(x) \right). \tag{2.1}$$

This error is also aptly called the *generalization error* because it is related to how the model would perform in the real world. Because $\mathcal{D}$ is not available to the learner, the generalization error can only be estimated from the error on the data we have: the training data $\mathcal{S}$.

One reasonable approach would be to minimize the training error over $\mathcal{S}$. So realistically, after learning and training, the learner outputs a classifier $\hat{h}_{\mathcal{S}} : \mathcal{X} \to \mathcal{Y}$ that is dependent on $\mathcal{S}$ (hence the subscript) and that minimizes the training error

$$L_{\mathcal{S}}(h) \triangleq \frac{1}{m} \sum_{i=0}^{m} \mathbb{1}_{h(x_i) \neq y_i} \tag{2.2}$$

Where $\mathbb{1}_{\text{condition}}$ is 1 if the condition is true and 0 if it is false. This error is also called the *empirical error*, and this paradigm of learning is subsequently termed *Empirical Risk Minimization* (ERM).

While minimizing training error seems like a good approach, it might not be enough to minimize generalization error. For instance, what if the learner simply memorized the data? The pathological classifier could technically return the correct output for a memorized input and flip a coin for inputs form outside the training data. In this case, the classifier would achieve 100% training accuracy while predicting very poorly in the real world: it would not generalize well at all. This situation is suitably called *overfitting* and it is a prime example of the potential divergence between generalization and training errors.

It turns out there is a way to prevent overfitting by restricting the search space to a set of hypotheses called the *hypothesis class* $\mathcal{H}$. Classifiers $h \in \mathcal{H}$ will share a common characterisitc, such as functional form; the set of all possible lines of best fit is a hypothesis class, the set of all possible feed-forward neural networks with some number of hidden layers and some combination of nodes in each layer is another hypothesis class. Given $\mathcal{H}$ and $\mathcal{S}$, the learner will find a predictor $h \in \mathcal{H}$ which miminizes the training error:

$$\text{ERM}_{\mathcal{H}}(\mathcal{S}) \in \operatorname*{arg\,min}_{h \in \mathcal{H}} L_{\mathcal{S}}(h). \tag{2.3}$$

Choosing a more restricted hypothesis class will prevent overfitting better, but there is a tradeoff. Deciding what $\mathcal{H}$ constitues for a particular problem means that we have to encode some prior knowledge that we have about how the data was generated in the first place, and this will introduce a specific form of bias called *inductive bias*. In summary, if our model is not restricted enough, it might overfit, and if it is too restricted, it will be biased. This observation is known as the *bias-complexity trade-off*: a hypothesis class $\mathcal{H}$ that is very rich decreases the approximation error while increasing the estimation error resulting from overfitting, whereas a smaller $\mathcal{H}$ reduces overfitting error while increasing approximation error.

This "richness" characteristic can be formally understood by the sample complexity within the probably approximately correct (PAC) learning framework. PAC learning is essentially a more realistic learning framework that introduces an accuracy threshold ϵ for what we consider to be a 'good' accuracy and the probability δ that our learner finds a 'bad' hypothesis in a given class $\mathcal{H}$. The 'probably' and 'approximately correct' in PAC refer to δ and ϵ respectively. The PAC learning framework assigns for each hypothesis class a function (e.g. the sample complexity) $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ of $(\delta, \epsilon) \in (0,1)^2$ which sets a lower bound for the number of i.i.d. data points needed to find a hypothesis h such that when $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, with probability of at least $1 - \delta$, the hypothesis is going to be approximately correct: $L_{\mathcal{D},f}(h) \leq \epsilon$.

Learning theory studies how complex we can make $\mathcal{H}$ while still achieving reasonable error. For finite hypothesis classes (which we do not encounter in this thesis), it turns out that sample complexity is proportional to the size of the class $|\mathcal{H}|$:

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil \tag{2.4}$$

So the larger the hypothesis class is, the larger our training data should be to avoid overfitting. For infinite hypothesis classes, the Vapnik-Chervonenkis dimension (VC-dimension) is used to compute the sample complexity instead of $|\mathcal{H}|$. The VC-dimension of a hypothesis class is the maximum number of data points such that a hypothesis from $\mathcal{H}$ can predict all combinations of labels on these data points.

A key takeaway here is the importance of the sample complexity of a hypothesis class is in determining its ability to generalize. A plausible rule of thumb is that the complexity of our predictors should match to a reasonable extent the complexity of the generative models that result in the distribution D over all the possible inputs. Otherwise, a lot more data will be required. This is why deep neural networks (DNNs) are not used when a simple linear regression would do; a DNN may overfit the data and return nonsense for inputs outside its training set, unless it is given a lot more data (or certain techniques to regularize or constrain the model are employed). This is also why linear regression is not used to classify images, its low sample complexity will increase the bias and therefore the estimation error. Figure **2.1** demonstrates this rule of thumb visually: the hypothesis classes should loosely match the complexity of the generative models that produce the genomic sequences we are trying to classify.

Another important and often implicit assumption is the i.i.d. assumption; data that is independent and representative of $\mathcal{D}$ is essential to finding generalizable classifiers. One can make the connection between the i.i.d.

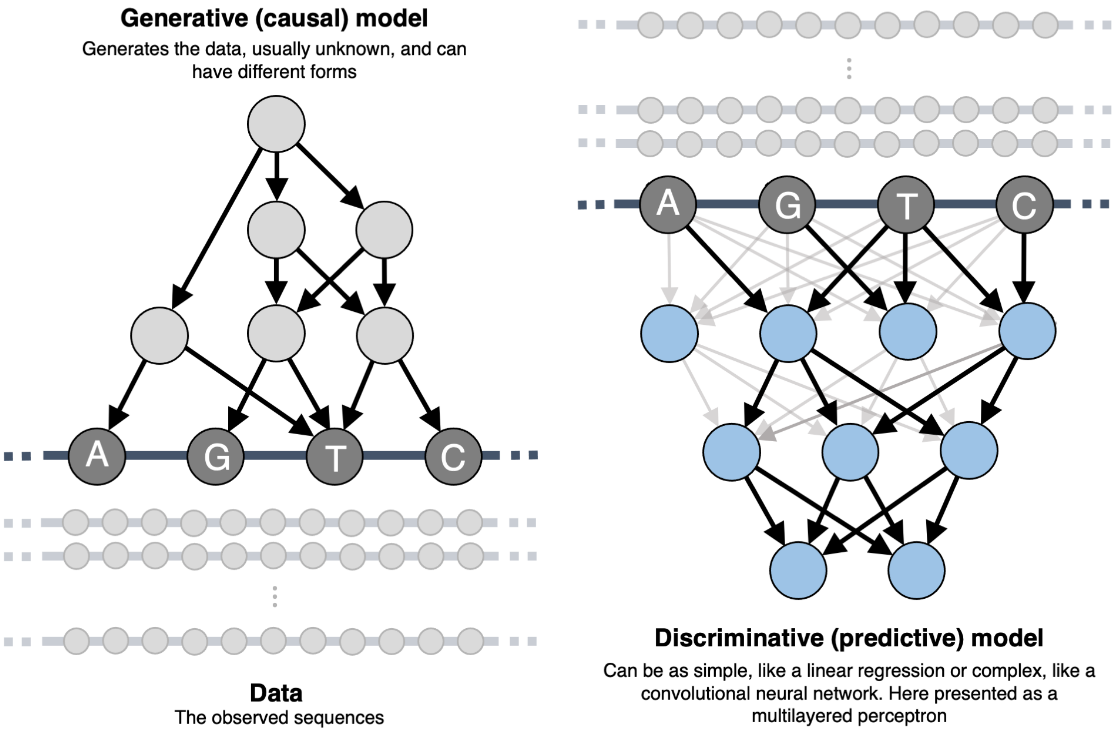

Generative (causal) model
Generates the data, usually unknown, and can have different forms
A
G
T
C
Data
The observed sequences
A
G
T
C
Discriminative (predictive) model
Can be as simple, like a linear regression or complex, like a convolutional neural network. Here presented as a multilayered perceptron

Figure 2.1: **Prediction as going from generative to discriminative models.** One can think of a generative model as a sequence of samplings that go from the top of the causal directed acyclic graph (DAG) to the bottom of it, where data is observed. The top node could represent whether there is fire in the building and the bottom (dark grey) nodes could represent whether there is smoke or whether the fire alarms are going off. One can think of a discriminative model as any method of prediction we could use. It could be as simple as a rule of thumb ("there is no smoke without fire"), a linear or logistic regression, or something as complex as a deep neural network and more. After seeing a lot of *labeled* data (labeled as in we know whether there is fire or not for every observation), we learn to predict the hidden variables. The graph on the right shows what happens after the learning process; notice how the learned network reflects the generative one. Although neural networks are very rarely that human-interpretable, they perform well. This example is mainly to illustrate the importance of having a discriminative model that can accommodate the complexity of the generative processes that produce the data.

assumption and the diversity of the sequences used to estimate sequence conservation measures. Sequences that are too homologous or not homologous enough are unrepresentative of the real-world distribution $\mathcal{D}$, and will therefor lead to poor generalizability. The concepts of model complexity and data distributions will be key to our understanding of the methods elaborated in this thesis.

2.2 The Roadmap

Figure **2.2** shows the roadmap for the rest of this chapter. Though there is a plethora of computational methods used for prediction, they can be categorized according to at least two of their aspects: the 'complexity' of the model

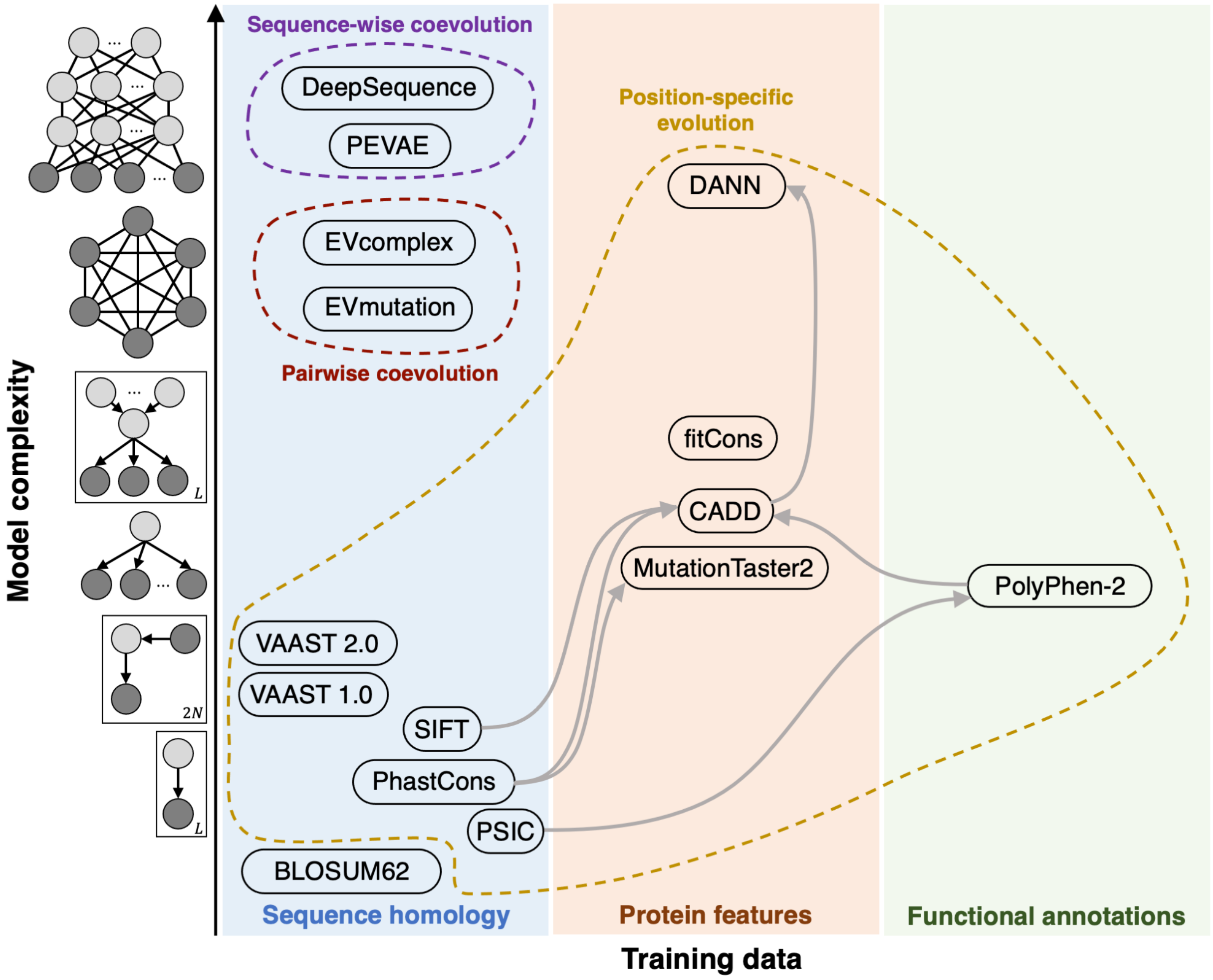


Figure 2.2: **A roadmap of pathogenicity prediction methods.** It is useful to dissect the overarching frameworks of the variant pathogenicity prediction methods that have been developed so far. When thinking about such methods, it is useful to known *what* types of features or data is being used (as seen on the categorical x-axis) and *how* these features or data are being used. One way to classify these models is according to increasing sample complexity of their hypothesis class, as seen on the y-axis. With the emergence of deep learning algorithms, prediction models have gotten more complex; from simple frequentist statistical schemes to models that attempt to approximate the intricate ways in which sequences evolve. BLOSUM: Blocks Substitution Matrix; SIFT: Sort Intolerant from Intolerant; VAAST: Variant Analysis, Annotation, and Search Tool; CADD: Combined Annotation-Dependent Depletion; DANN: Deleterious Annotation of genetic variants using Neural Networks; PEVAE: Protein Evolution Variational Autoencoder; fitCons: fitness consequence; PSIC: Protein Specific Independent Counts.

(e.g. the hypothesis class) and the types of data used. Model complexity here is mostly qualitative; it depends on the the number of parameters and the structure of the generative graph model that is implied by a method (see the plate notation on the vertical axis of the Figure 2.2). As we saw in Section 1.3, there are more than just three types of data. However, one could think of allele frequencies as data extracted from large homologous sequence databases. Other useful aspects for categorization include a model's assumptions about coevolution and epistasis; some methods assume that positions evolve independently, pairwise, or sequence-wise. We will elaborate on all of this later.

One can notice many things by looking at this roadmap. First, most methods explicitly or implicitly make simplifying assumptions about the nature of sequence evolution, and this correlates well with low model complexity. Second, we notice that the most complex methods (DeepSequence, PEVAE, EVmutation, and EVcomplex) all capitalize on sequence conservation. Indeed, as we will see later, there are many approaches to extract evolutionary information from multiple sequence alignments, and these approaches assume rather complex underlying structures for the data. More generally, we observe that the same type of data can be used in many different ways. This realization is a major one, as it re-emphasizes the divergence between underlying generative models and discriminative (predictive) hypothesis classes, the former of which may be a lot more entangled than the latter.

We will be reviewing these methods starting at the bottom of Figure 2.2 and in order of increasing model complexity. Most of these methods rely on more general computational techniques that were originally created for purposes other than variant pathogenicity prediction; we will briefly review those too.

2.3 Starting simple with BLOSUM62

Blocks Substitution Matrix (BLOSUM) is a family of substitution matrices used to score the alignment of protein sequences (Henikoff and Henikoff, 1992). These matrices use the log-odds score of a particular amino acid substitution to find the best alignment between two or more sequences.

The log-odds scores are calculated from thousands of *blocks* of sequences (hence the name). Each of these blocks is a gap-less local alignment of s protein sequences of length (width) w. The number 62 in BLOSUM62 refers to the minimum identity score (w minus the Hamming distance) between a sequence and at least one other sequence in the alignment. Therefore, BLOSUM45 would ideally score distantly related proteins, BLOSUM80 would score closely related proteins, and BLOSUM62 would score proteins more related than those scored by BLOSUM45 and less related than those scored by BLOSUM80.

To compute a log-odds score for a particular block, let n_a to be the number of times amino acid $a \in \{1, \ldots, 20\}$ appears at a certain position. Because a substitution could have happened from any sequence to the other, we will consider every substitution possible; this corresponds to setting every sequence as the query sequence and then counting the number of substitutions n_{ab} in the position between the query amino acid a and the aligned amino acid b. Using simple combinatorics, we get:

$$n_{ab} = \begin{cases} \binom{n_a}{2} & a = b \\ n_a n_b & a \neq b \end{cases} \tag{2.5}$$

These counts can be grouped in 20×20 substitution matrix. The log-odds score is then defined by Henikoff and Henikoff as

$$\text{log-odds score} = \log_2 \left(\frac{p_{ab}}{e_{ab}} \right), \tag{2.6}$$

where $p_{ab} = n_{ab} / \sum_{x=1}^{20} \sum_{y=1}^{x} n_{x,y}$ is the proportion of $a \to b$ transitions, and e_{ab} is the probability of finding a and b independently in the block (p_a^2 if $a = b$ and $2p_a p_b$ if $a \neq b$).

The BLOSUM62 matrix follows our intuition about the physical and chemical properties of different amino acids: log-odds scores are usually positive for similar amino acids and negative for dissimilar ones. For instance, substitutions between leucine and isoleucine have positive log-odds while substitutions between leucine and aspartate have negative ones.

One of the earliest computational methods for variant interpretation was used in a study of disease-related non-synonymous (coding) SNPs (Cargill et al., 1999). The classifier used then was strikingly simple: substitutions having a non-negative entry in the BLOSUM62 matrix were classified as conservative (tolerated), whereas substitutions having a negative entry in the BLOSUM62 matrix were classified as non-conservative (non-tolerated).

Of course, there are many issues with this idea. Substitution-scoring matrices such as BLOSUM62 are used for basic alignment search and are computed from large aggregated blocks of sequences, they do not integrate any position-specific information to the protein sequence of interest. The majority of the entries in the BLOSUM62 matrix are negative, which makes the method inherently overestimate the deleteriousness of an input variant. Selective pressure and conservation usually vary from one protein to another and within the many domains of a single protein, whereas the BLOSUM approach invariably considers every sequence to be subjected to the same level of

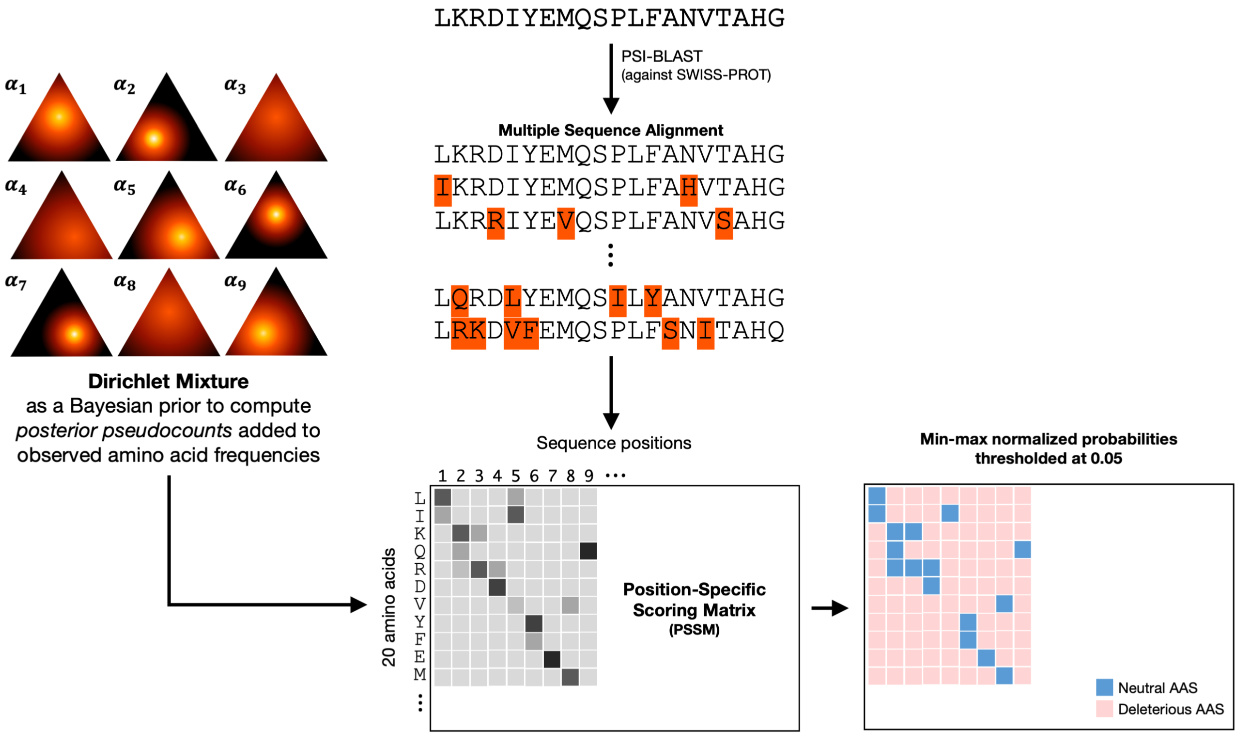


Figure 2.3: **A concept map of SIFT.** SIFT is one of the earliest methods developed to predict the effect of missense mutations in proteins. After a query sequence is submitted, the method proceeds as follows: (1) find homologous sequences via `PSI-BLAST` and build a multiple sequence alignment (MSA) from these sequences, (2) construct a position-specific substitution matrix (PSSM) of substitution frequencies, (3) compute the posterior frequencies using a mixture Dirichlet prior and update the PSSM with these frequencies, (4) min-max normalize the probabilities and threshold at 0.05. Substitutions with probabilities below this threshold are deemed to be deleterious.

evolutionary pressure.

2.4 Position-Specific Models of Homology and Structure

The main issue with the BLOSUM62 approach from Cargill et al. is its neglect of rather obvious position-specific information. On the other hand, all the upcoming methods use information that is specific to each position in the genome.

2.4.1 SIFT

The first type of position-specific data used was sequence conservation information, which was the central aspect of early methods like Sorts Intolerant From Tolerant (SIFT). SIFT's rationale is simple: tolerance to an amino acid at a particular position is proportional to its frequency at this position in a population of sequences (Ng and Henikoff, 2006). This follows from our understanding of purifying selection: deleterious alleles are differentially removed from a population.

Given a query sequence, SIFT uses `PSI-BLAST` to search for homologous sequences in the protein database SWISS-PROT. As we saw in Section **2.1**, it is important for those sequences to be representative of the true population to ensure the method is generalizable. To do this, the method uses a conservation scheme that adds sequences to the alignment until conservation decreases. SIFT defines conservation at a position i using the Shannon entropy of the empirical distribution of amino acids at i:

$$c_i = \log_2(20) - \sum_{a \in \{1,\ldots,20\}} f_{i,a} \log_2(f_{i,a}) \tag{2.7}$$

Where $f_{i,a}$ is the observed frequency of amino acid a at position i. This way, c_i is 1 when only one amino acid is observed (e.g. the position is conserved) and 0 when all amino acids are observed uniformly (e.g. the position is not conserved). The final sequence alignment is grown via cycles of adding sequences that only increase the sequence-wide conservation measure $\sum_i c_i$.

Once the alignment routine is completed, a position-specific substitution matrix (PSSM) is constructed using the observed frequencies and the posteriors from a 13-component mixture Dirichlet posterior. The Dirichlet mixture simply adjusts the observed frequencies according to our prior beliefs about the position-specific amino acid distribution and the observed data. Each entry $p_{i,a}$ in the PSSM is simply a weighted sum of the observed frequency and the Dirichlet posterior (interpreted as a pseudocount).

To account for the fact that the maximum probability at any position can be as low as $1/20$ (in the case where all amino acids are equally likely), the position-specific distributions are min-max normalized to yield new distributions such that $\tilde{p}_{i,a} = p_{i,a} / \max_a(p_{i,a})$. Variants with $\tilde{p}_{i,a} \geq 0.05$ are predicted to be benign, whereas variants with $\tilde{p}_{i,a} < 0.05$ are predicted to be deleterious (see Figure **2.3** for a brief visual overview of SIFT).

Ng and Henikoff compare the performance of SIFT to that of BLOSUM62 by applying them to predict the effects of mutations in three controlled mutagenesis studies with thousands of mutations each. The results of this comparison are expected: SIFT consistently outperformed BLOSUM62, but the accuracies were nonetheless quite lackluster, with a range of 63-81% for SIFT and 47-70% for BLOSUM62. In addition, BLOSUM62's deleterious prediction accuracy exceeded that of SIFT significantly on 2/3 data sets, whereas SIFT's tolerant prediction accuracy exceeded that of BLOSUM62 on all data sets. This last observation is in agreement with our earlier statement about BLOSUM62 being too conservative because it overestimates the deleteriousness of amino acid substitutions.

2.4.2 Polyphen-2

Polyphen-2 is another model commonly used in pathogenicity prediction programs (Adzhubei et al., 2010). It is one of the first methods to use structural information collected from protein data bank (PDB) folds and to combine more than just one feature for prediction. It is a Naïve Bayes algorithm that estimates the probability that a missense mutation is damaging to the function of a protein, and it uses 11 features to estimate that probability: 8 of which are sequence conservation-based and the remaining 3 being structure-based. Table **2.1** lists all 11 features with a short description for each feature.

Naïve Bayes classifiers are a family of simple probabilistic classifiers that make strong assumptions of independence across features and apply Bayes' theorem to compute the conditional probability that a variant is deleterious given its feature vector. From Bayes' theorem:

$$\begin{aligned} P(Y=1 \mid \boldsymbol{X}=\boldsymbol{x}) &= \frac{P(\boldsymbol{X}=\boldsymbol{x} \mid Y=1) \cdot P(Y=1)}{P(\boldsymbol{X}=\boldsymbol{x})} && (2.8) \\ &\propto P(\boldsymbol{X}=\boldsymbol{x} \mid Y=1) \cdot P(Y=1) && (2.9) \end{aligned}$$

Where $Y = 1$ is the event that a variant is deleterious and $\boldsymbol{x} = (x_1, \ldots, x_{11})$ is the vector of 11 features associated with that variant. Realistically, the features in $\boldsymbol{x}$ are not independent of each other:

$$P(\boldsymbol{X}=\boldsymbol{x} \mid Y=1) = P(x_1 \mid Y=1) \cdot P(x_2 \mid x_1, Y=1) \cdot \cdots \cdot P(x_{11} \mid x_1, \ldots, x_{10}, Y=1), \qquad (2.10)$$

which can be quite unwieldy. However, we can make the 'Naïve' Bayes assumption of independence among the different features to get a much simpler result:

$$P(\boldsymbol{X}=\boldsymbol{x} \mid Y=1) = \prod_i P(x_i \mid Y=1) \qquad (2.11)$$

This allows us to rewrite the posterior as

$$P(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \propto \prod_i P(x_i \mid Y = 1) \cdot P(Y = 1) \tag{2.12}$$

$$= \frac{1}{Z} \prod_i P(x_i \mid Y = 1) \cdot P(Y = 1) \tag{2.13}$$

where $Z = \sum_{y \in \{0,1\}} \prod_i P(x_i \mid Y = y) \cdot P(Y = y)$ is a normalizing constant. The conditional probabilities $P(x_i|D)$ are conventionally estimated using entropy-based discretization followed by Laplace additive smoothing. (See **??** and **??** for further information on entropy-based discretization and Laplace smoothing, respectively.)

For training, Polyphen-2 uses two pairs of labeled data sets. The first, HumDiv, uses damaging alleles annotated in the UniProt database together with alleles from closely related mammalian homologs considered to be non-damaging. The second set, HumVar, uses all the human disease-causing mutations found in UniProt and assumes non-synonymous SNPs which are not annotated as disease-causing to be non-damaging.

Polyphen-2 works well and is a major common example for methods used in variant interpretation and prioritization, since deleteriousness is strongly correlated with the loss of function of a protein, which in turn correlates with higher variant pathogenicity. With a false positive rate of 20%, the algorithm reaches an accuracy of up to 92% on the HumDiv data set and 73% on the HumVar data set.

However, besides its non-trivial false positive rate, it does have shortcomings that it shares with SIFT, BLOSUM62, and most methods of similar complexity. All these approaches can only take missense coding mutations as input, which is a major limitation since mutations do not always appear in the translated gene product. They do not cover non-coding regions, and they do not cover insertion-deletions. Another key observation is that they are meant to estimate the deleteriousness of a variant — not its pathogenicity. As we saw in the introduction to this thesis: damaging does not necessarily mean pathogenic, and pathogenic does not necessarily mean damaging (Eilbeck et al., 2017). All these issues are important and will be periodically revisited throughout this thesis.

2.4.3 VAAST 2.0

Like SIFT and Polyphen-2, the Variant Annotation, Analysis, and Search Tool (VAAST) 2.0 harnesses the predictive power of sequence conservation (Hu et al., 2013). However, instead of only incorporating the frequencies of amino acids at a particular position, VAAST 2.0 also includes in its analysis dichotomous labels (e.g. 'affected' versus 'unaffected') for every sequence as well as conservation probability estimates from PhastCons. VAAST 2.0

Feature	Description
Sequence and Conservation Features independent of tertiary-structure	
PSIC score	Position-specific independent counts: position-specific probability of observing the wild-type amino acid at the mutation site considered. Computed from counting and weighting amino acids found in a multiple sequence alignment of the gene variant.
ΔPSIC score	Difference in PSIC scores between the wild-type and mutant residues. Large positive values mean the mutant is more likely to be observed, large negative values mean the mutant is less likely to be observed.
Alignment depth	The number of residues observed at the position across the multiple alignment, excluding gaps.
Maximum congruency	The congruency of the mutant allele to the multiple alignment: for each of the 20 amino acids, the sequence identity between the analyzed protein and its closest homologue wherein this amino acid is observed is computed. The identity percentages are then weighted by the BLOSUM matrix, and the maximum weighted value is returned.
Maximum identity	Sequence identity with the closest homologue deviating from the wild-type allele. Can be thought of as a proxy to how 'unique' the protein is.
CpG context	Whether the variant happened as a transition or transversion in a CpG context, or neither.
Pfam hit	Whether the position of the mutation is within or outside a protein domain defined by Pfam.
Volume change	The change in volume between the wild-type and the mutant residue side chain.

Structural Features for proteins with known tertiary structures	
B-factor	Crystallographic temperature B-factor (also known as the Debye-Waller factor).
Accessible surface	The normalized accessible surface of the wild-type amino acid residue.
propensity	The change is accessible hydrophobic surface area propensity for buried residues, estimated using knowledge-based potentials.

Table 2.1: The list of features used in Polyphen-2 and their respective descriptions.

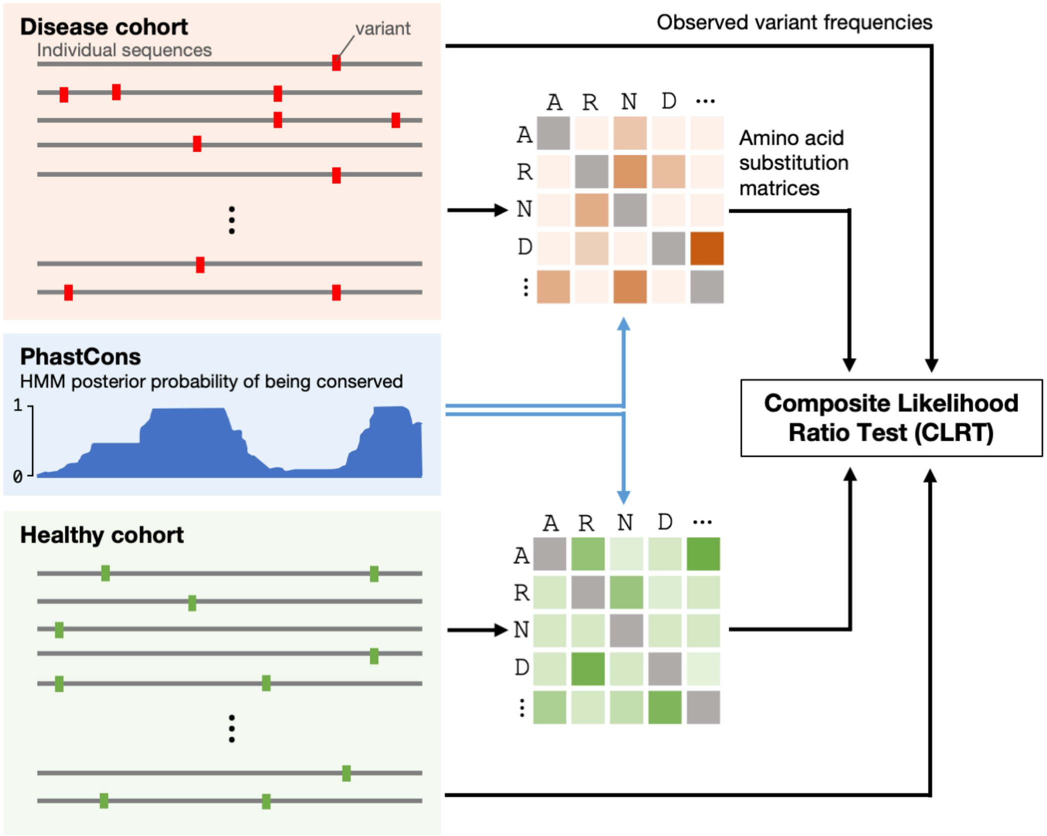

Disease cohort
Individual sequences
variant
Observed variant frequencies
A R N D ···
A
R
N
D
⋮
Amino acid substitution matrices
PhastCons
HMM posterior probability of being conserved
1
0
Composite Likelihood Ratio Test (CLRT)
Healthy cohort
A R N D ···
A
R
N
D
⋮

FIGURE 2.4: **A concept map of VAAST 2.0.** VAAST 2.0, like many other tools used for genome wide association studies (GWAS, which have also been used for variant prioritization), relies on two sequence data set: one from a healthy cohort and one from an affected one. Using these sequences and the substitution matrices calculated from them, the collapsing method, and probability estimates from PhastCons, VAAST 2.0 decides whether a variant is pathogenic or not based on a composite likelihood ratio test.

is similar to other GWAS disease-gene association methods in that it uses two sequence data sets (one from affected individuals and one from healthy individuals) and a statistical test to check whether a variant is differentially expressed in one or the other. The score output is the p-value of a composite likelihood ratio test. This similarity to GWAS methods makes VAAST 2.0 capable of scoring single variants, genes, or genomic loci.

The setup is the following. VAAST 2.0 takes as input two alignment blocks; one with $n^{(U)}$ sequences from unaffected individuals and the other with $n^{(A)}$ sequences from affected individuals. Let $i = 1, \ldots, m$ index all the variants observed in these sequences, and let $X_i^{(U)}$ and $X_i^{(A)}$ denote the number of times variant i is observed among the unaffected and affected groups, respectively. From the binomial distribution, we know that

$$X_i^{(U)} \overset{\text{i.i.d.}}{\sim} \text{Bin}\left(p_i^{(U)}, n^{(U)}\right), X_i^{(A)} \overset{\text{i.i.d.}}{\sim} \text{Bin}\left(p_i^{(A)}, n^{(A)}\right) \tag{2.14}$$

The null hypothesis stipulates that both $X_i^{(U)}$ and $X_i^{(A)}$ are distributed with the same binomial distribution, that is, $p_i^{(U)} = p_i^{(A)}$, whereas the alternative hypothesis is that the probabilities are different $p_i^{(U)} \neq p_i^{(A)}$. From this, one easily gets the likelihood ratio test for many possible variants in one gene or genomic locus:

$$\lambda = \sum_{i \in [m]} \log \left(\frac{\hat{p}_i^{X_i} (1 - \hat{p}_i)^{n_i - X_i}}{\left(\hat{p}_i^{(U)}\right)^{X_i^{(U)}} \left(1 - \hat{p}_i^{(U)}\right)^{n_i^{(U)} - X_i^{(U)}} \left(\hat{p}_i^{(A)}\right)^{X_i^{(A)}} \left(1 - \hat{p}_i^{(A)}\right)^{n_i^{(A)} - X_i^{(A)}}} \right) \tag{2.15}$$

Where $X_i = X_i^{(U)} + X_i^{(A)}$ and $n_i = n_i^{(U)} + n_i^{(A)}$. This likelihood ratio test (LRT) is augmented using a few heuristics. First, from each sequence block, a substitution matrix is created, and a ratio of substitution probabilities is used in the LRT to account for the severity (or lack thereof) of an amino acid substitution. Additionally, PhastCons, an HMM-based model that estimates the posterior probability that a position in the genome is conserved, is used too in the LRT. Once the scores are computed, a permutation test with the sequences is performed to return p-values for gene and variant prioritization. Hu et al. compare VAAST 2.0 to several other variant prioritization tools such as Polyphen-2, SIFT, MutationTaster, and an array of GWAS models, and VAAST consistenly outperforms most of them by varying margins.

In the case of rare variants and indels, VAAST 2.0 does what is usually done in GWAS methods of its generation: it collapses the variants. This means that a small locus with several rare variants is considered as one position in the genome. The collapsing trick is useful but not realistic, as it ignores position-specific information that might still be important, even if it is rare. A main weakness of this method is that it only works for variants with dichotomous traits (healthy versus disease) and cannot generalize to continuous phenotypes.

2.4.4 CADD

To address the absence of a framework for variant interpretation, Kircher et al. propose an interesting idea: in a mechanism similar to survivorship bias, variants that are observed in the general population tend to be benign whereas deleterious variants are depleted in the population by natural selection (Kircher et al., 2014). Therefore, deleterious variation will be enriched in simulated and uniformly random variation variants, whereas benign variation will be enriched in the observed variants of a population. Combined Annotation-Dependent Depletion (CADD) is a method that relies on this idea (Kircher et al., 2014).

A newly discovered variant that primarily resembles simulated variants is more likely to be deleterious than a

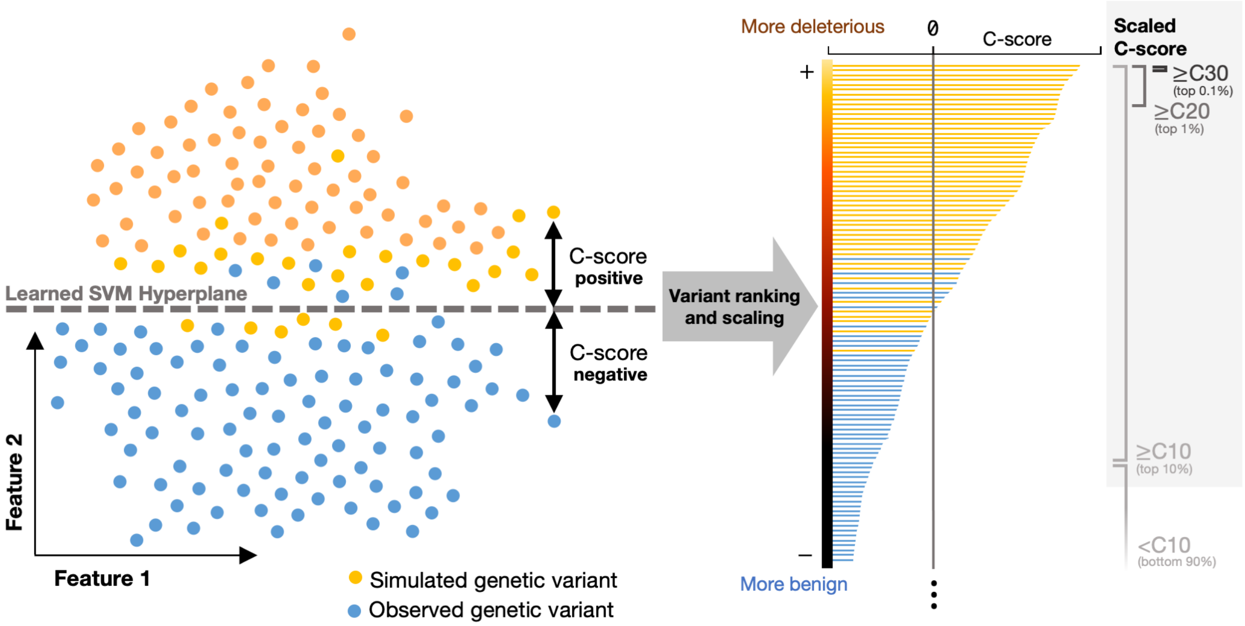


FIGURE 2.5: **A concept map of CADD.** CADD uses an original framework for mutation effect prediction. It considers all observed variants as benign ones (having survived purifying selection), and all simulated variants as deleterious (having not been seen in the population). CADD uses 63 annotations (encoded into 949 features) and learns an linear kernel SVM classifier on these features. The C-score is simply the distance between the variant and the SVM hyperplane, and the more positive it is, the more simulated (e.g. deleterious) the variant is expected to be. The rightmost part of this figure shows how the *scaled* C-scores are computed. After the variants are ranked, the top 10% get C-scores of C10 and more, the top 1% get C20, the top 0.1% get C30, and so on.

variant that primarily resembles observed variants. This way, we can compare the relative deleteriousness of different variants by simply ranking them according to how similar they are to variants generated by a large simulation of human genetic variation. This novel framework supposedly overcomes the ascertainment bias that might be seen in curated variant data sets simply because it does not rely on any manual estimates of pathogenicity (which can be biased and unreliable, as explained above).

CADD brings the feature-savviness of Polyphen-2 to a new level. Instead of 11 features, it uses 63 found across several annotation projects including the Ensembl Variant Effect Predictor (VEP), the ENCODE Project (Dunham et al., 2012), and the UCSC Genome Browser tracks (Meyer et al., 2013). For model training, those 63 features were encoded into 949. They include conservation metrics, regulatory information, transcription factor binding, transcript information, and protein-level and structural information, and even the estimates of other classifiers such as Polyphen-2 itself, GERP, SIFT, PhastCons, PyloP, and many others. The classifier is a linear kernel Support Vector Machine (SVM) trained on over 13 million simulated single nucleotide variants, 627,000 simulated insertions, and 926,000 simulated deletions (CADD also supports insertion-deletions). The output is a proxy for the probability that a variant is simulated — and therefore deleterious.

The ultimate result is a scaled C-score that is high for simulated variants (expected to be deleterious) and low for observed ones (expected to be benign). Kircher et al. show that the higher a variant's scaled C-score is, the lower its frequency is in a population according to the 1000 Genomes Project data set. They also show that observed variants associated with Mendelian diseases tend to have higher scaled C-score than variants that are not associated with disease.

Despite the many improvements CADD brings to variant prioritization and interpretation, it suffers from limitations too. One such limitation is the lack of gold-standard data in non-coding regions of the genome, which reduces accuracy significantly in those regions. C-scores are correlated with variant deleteriousness but they are also affected by other factors that CADD does not account for (i.e. mutation rates, differences in selective intensity, background selection, and others). These biases can potentially reduce accuracy too. There is also no direct relationship between CADD-estimated deleteriousness and the pathogenicity of a variant, since it returns a *scaled* score that only has meaning in the context of the scaled C-scores of other variants.

Furthermore, though CADD brings a novel perspective to variant prioritization by comparing variants to observed and simulated ones, this model is not rooted in nor inspired by a coherent biological model. CADD leaves significant further room for development. How would other (non-SVM) models compare? Would non-linear kernels perform better?

2.4.5 Going Deep with DANN

Deleterious annotation of genetic variants using neural networks (DANN) is CADD, but with a deep neural network (DNN) instead of an SVM as predictive model. It uses the same exact features and data that CADD uses to train a DNN with 3 hidden layers of 1000 nodes with hyperbolic tangent activation functions (see Figure **2.6**). The loss function used is cross entropy and the learning algorithm is Stochastic Gradient Descent with momentum (see **??** and **??** for an elaboration of these topics). To avoid overfitting, DANN uses a hidden node droupout rate of $p = 0.1$. Dropout in deep learning is a powerful trick wherein for every training cycle, each of the nodes in a neural network is 'dropped out' with probability p; that is, it is ignored completely by the network. This strategy roughly doubles the time it takes for a DNN to converge, but it significantly improved testing accuracy (generalization). It is thought that droupout makes the features learned by the hidden nodes more robust.

The rationale behind replacing a linear kernel SVM with a DNN is that DNNs account for non-linearities,

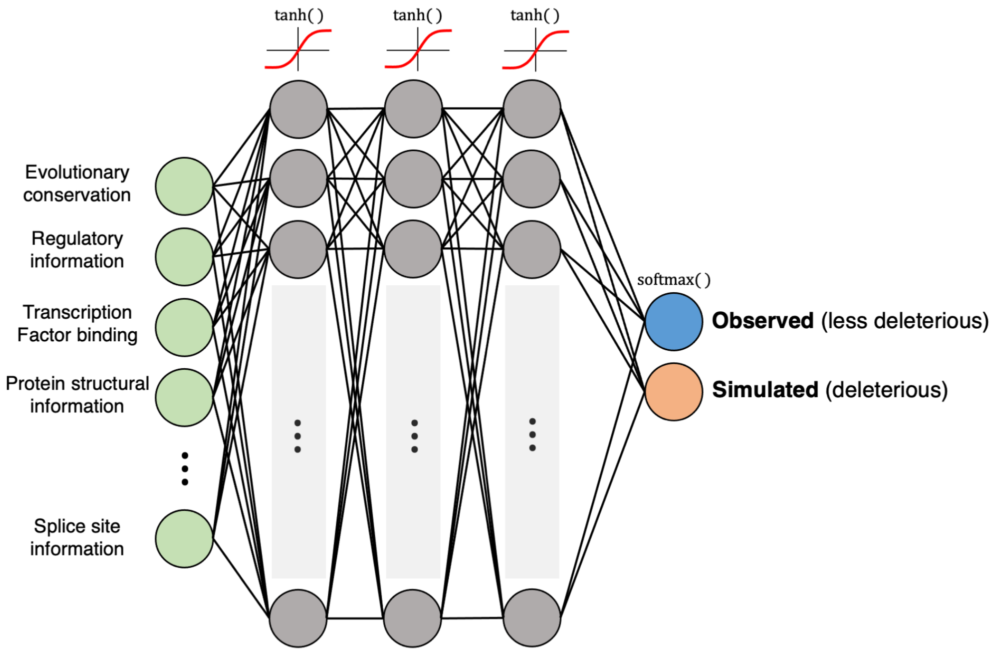

tanh( )
tanh( )
tanh( )
Evolutionary conservation
Regulatory information
Transcription Factor binding
Protein structural information
Splice site information
softmax( )
Observed (less deleterious)
Simulated (deleterious)

Figure 2.6: **The architecture of DANN**. DANN uses the same annotations and features as CADD to classify whether a variant is observed or simulated. However, to account for non-linear relationships between a variant's features (which a linear kernel SVM cannot capture), DANN uses a deep neural network (DNN) composed of three 1000-node and hyperbolic tangent-activated fully connected layers. Surprisingly, and despite its depth and width, DANN only performs a little better than CADD.

which is a more realistic assumption about such convoluted data. And indeed, the DNN approach increases accuracy. However, both the SVM and DNN models have limited performances on the classification of variants as simulated versus observed: 58.2% and 66.1%, respectively. There are many reasons for why this could be, one of which may be the fact that the simulated/observed divide is not as aligned with the deleterious/benign one as one would think.

Further, DNNs are almost always a non-human interpretable black box, using them to model another black box such as variant pathogenicity substantially decreases their reliability.

2.4.6 fitCons

fitCons aims to estimate the 'fitness consequence' (hence the name) of a mutation at any position in the genome (Gulko et al., 2015). And like all of the methods we have discussed so far, is also based on sequence homology and conservation. As we have established already, conservation-based methods are contingent on a few assumptions of sequence independence and orthology which might not always be reasonable. To address these issues, one might

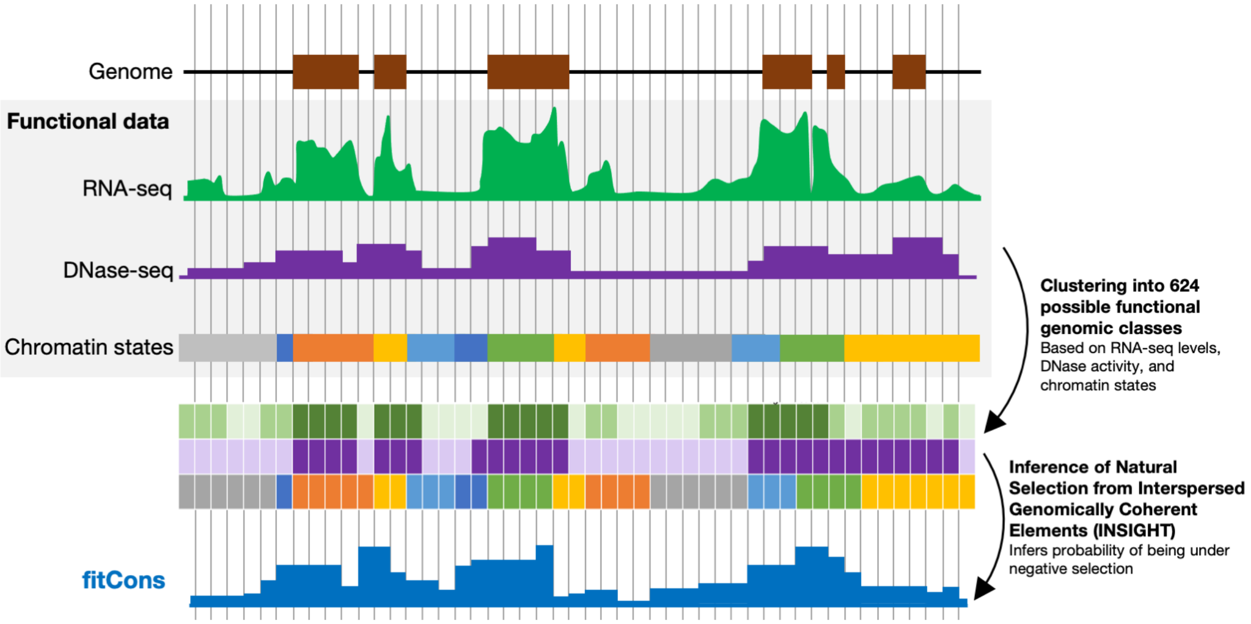

Genome
Functional data
RNA-seq
DNase-seq
Chromatin states
fitCons
Clustering into 624 possible functional genomic classes
Based on RNA-seq levels, DNase activity, and chromatin states
Inference of Natural Selection from Interspersed Genomically Coherent Elements (INSIGHT)
Infers probability of being under negative selection

FIGURE 2.7: **A concept map of fitCons.** To overcome the problem of sparse signal, fitCons clusters genomic positions according to 3 types of functional data: (1) RNA-seq as a measure of transcription activity, (2) DNase-seq as a measure of chromatin exposure, and (3) chromatin states as inferred from a separate HMM algorithm. There are 624 total possible combinations of values from all these measures, and thus the entire genome is divided into 624 partitions. After clustering, INSIGHT is used to estimate ρ, the probability that a site is under negative selection, for each site in the genome. Because INSIGHT is based on homology, divergence, and phylogeny, one can say that fitCons is based on these paradigms too.

either rely more on intraspecies variation (as opposed to interspecies) or simply do without evolutionry information and instead use functional genomic information alone. fitCons essentially combines these two strategies into one.

fitCons first clusters genomic regions according to functional fingerprints composed of three signals. The first is DNase-seq, which reveals genomic regions with open and accessible chromatin; the second is RNA-seq, which indicates the level of transcription of a particular locus; and chromatin immunoprecipitation and sequencing (ChIP-seq) data which describes chromatin states. These functional features of the genome are very useful, as they exist throughout the entire genome and they correlate well with evolutionary dynamics. For instance, proteins that are highly expressed (with high RNA-seq signal) tend to be under stronger selective pressure.

Once each genomic position is classified according to that scheme, the fraction of sites ρ under selection is computed using a tool called INSIGHT (inference of natural selection from interspersed genomically coherent elements) over all the genomic positions for each functional fingerprint category. Therefore, each cluster receives a value between 0 and 1 that represents the probability that this position influences fitness.

Because it relies on a lot of genome-wide accessible data, fitcons has an appreciably higher sensitivity and coverage in non-coding regions than almost all of the other methods compared. Additionally, different thresholds of the fitCons score demonstrated intuitive compositions of annotation types. For instance, low threshold were enriched for intronic and intergenic genomic positions (because most of the genome is intronic and intergenic), while high thresholds were enriched for positions in the coding sequence (CDS) of protein-coding genes (because they represent the most conserved positions in the genome).

While there is no direct link between fitCons and the deleteriousness of a given variant, it is expected that positions with high fitCons values will be less tolerant for mutations than positions with lower fitCons scores. Moreover, these scores could be used in future methods for variant pathogenicity prediction.

fitCons is a remarkable transition point in our roadmap of methods. It is a method that considers genomic positions to be dependent on each other, and it uses functional fingerprints in an attempt to understand this interdependence. All the upcoming method follow this new and profound paradigm of variant interpretation which gradually becomes more and more rooted in the biophysical and biochemical interactions that make proteins bind to DNA, RNA, other proteins, and other molecules altogether. These interactions are important because the energy landscapes they define are an essential component of the evolutionary landscapes of sequence populations.

2.5 Pairwise Evolutionary Couplings for Coevolving Sites

The *evolutionary couplings* (EC) framework is predicated on the profound observation that positions within the genome coevolve in sequence space, and that this coevolutionary trajectory is constrained by the function of these genomic positions. Using the clichéd 'cooking recipe' analogy of the central dogma — wherein DNA is the recipe, RNA is a copy of that recipe, and protein is the cooked food — coevolution can be likened to the process of simultaneously changing several parts of a recipe to improve it, knowing that only changing one part and not the other parts would deteriorate the resulting food.

Clearly, understanding these functional constraints would greatly help the development of tools for variant prediction and protein engineering. In fact, the EC framework was primarily established for *protein structure prediction* (PSP). The connection between EC and PSP is dependent on the assumption that evolutionary constraints correlate with the physical proximity of two residues in a protein fold, and indeed, experimental results strongly confirm the reasonableness of this assumption. The need to conserve (and improve) energetically favorable interactions leaves trailing patterns in the evolutionary record; the task at hand is to find those patterns in sets of homologous sequences and translate them into physical and probabilistic information useful for solving problems like PSP and

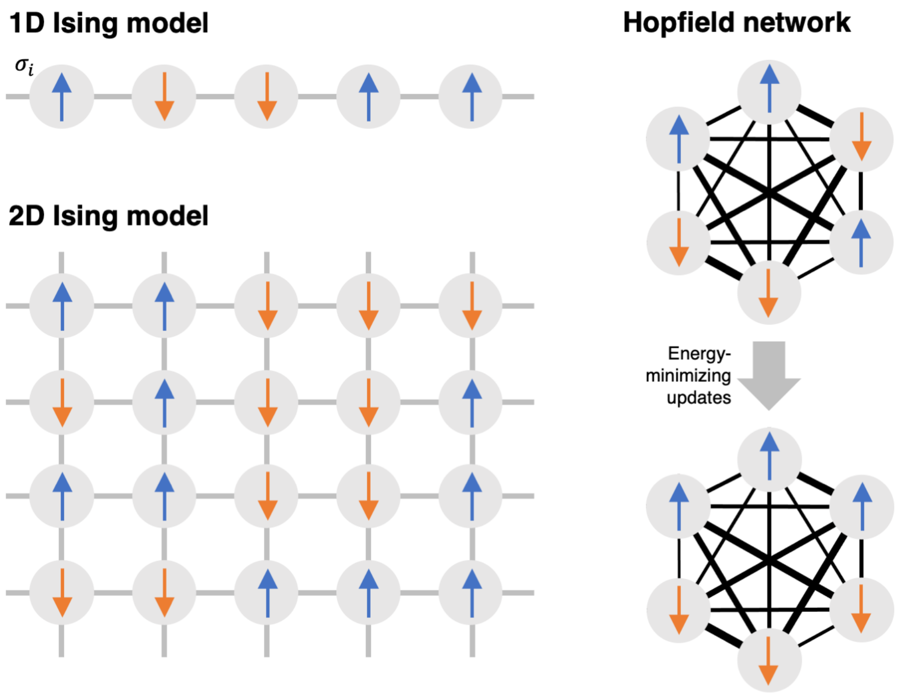


Figure 2.8: **Ising models and Hopfield networks.** Ising models are energy-based models used to simulate physical processes such as magnetization. The arrow inside each node represents the direction of the spin in that node, which can either be $+1$ or -1. A Hopfield network is an energy based model that was developed to simulate neural processes. Notice how the width of the connecting edges can vary; every combination of widths has its own energy landscape, and every landscape has its local and global minima. These minima are thought of as 'stored memories' because energy-minimizing updates will always converge to one of them when starting from any input.

variant pathogenicity prediction.

In summary, the EC framework has been used to predict residue contacts in protein folds (Marks et al., 2011; Ekeberg et al., 2013; Balakrishnan et al., 2011), approximate full protein 3D structures using the inferred proximity constraints (Hopf et al., 2012; Ovchinnikov et al., 2015), residue contacts at the interfaces of protein complexes (Hopf et al., 2014; Ovchinnikov et al., 2014), the effects of missense mutations (Hopf et al., 2017), and even more. The purpose of this section is to review the ways in which the EC framework allows one to accomplish all of the above. But first, a brief interlude is necessary to introduce energy-based models and capture the beautiful and intuitive connections to physics that make these models particularly appealing.

2.5.1 Ising, Potts, and Energy-based models

The Ising model is a mathematical model that has been used extensively to describe a multitude of physical phenomena, including ferromagnetism, phase transitions, lattice gases, metal alloys, and even 'bacterial vortex lattices' (Wioland et al., 2016). We will briefly describe the Ising model in the context of ferromagnetism to gain a physical intuition for it, and then generalize this intuition and apply it to sequence evolution.

An oversimplifying model for a magnet is a lattice of infinitesimally small mini-magnets, each of which has its own minuscule magnetic moment, which we will call 'spin'. The spin σ_i each lattice site i can either be -1 (for 'down') or $+1$ (for 'up'). This lattice can have any number of dimensions, but things get significantly more complicated as we go beyond the 1D Ising model. Only when the majority of the spins are aligned, the ordered lattice becomes a magnet with its own macroscopic magnetic moment. Otherwise, it's a disordered mess. A phase transition is the transition of the lattice between an ordered phase and a disordered phase, and where this transition occurs depends on temperature and on the energy of the lattice itself.

Speaking of energy, the Ising model's Hamiltonian is a function of two types of energy. The first type is the *interaction energy* J between the different lattice sites. Because each site can be thought of as a mini-magnet, the mini-magnets can feel each other's magnetic field, and this interaction affects the total energy of a particular microstate. The second type of energy is the *external field* h at each lattice site. (Notice h is not a hypothesis from learning theory anymore.) Without an external field, the configurations $+1$ and -1 are symmetric: they are equal in energy, and so an isolated lattice site is equally likely to be found in either state. When an external field is introduced, this symmetry breaks: a lattice site might favor pointing up if the external field is pointing up too. Taking these two energies into account, the Hamiltonian for a microstate $\boldsymbol{\sigma}$ becomes:

$$H(\boldsymbol{\sigma}) = -\sum_{\langle ij \rangle} J\sigma_i\sigma_j - \sum_{i=1}^{N} h\sigma_i \tag{2.16}$$

Where $\sum_{\langle ij \rangle}$ represents the sum over neighboring lattice site pairs. We can then find the probability of any microstate by using the Boltzmann distribution:

$$P(\boldsymbol{\sigma}) = \frac{1}{Z(J,h)} \exp\{-\beta H(\boldsymbol{\sigma})\} \tag{2.17}$$

Where $\beta = 1/k_bT$ is the inverse temperature and $Z(J,h) = \sum\limits_{\boldsymbol{\sigma}\in\{-1,+1\}^N} \exp\{-\beta H(\boldsymbol{\sigma})\}$ is the partition function, which is a normalizing sum over all possible N-long combinations of spins (denoted by $\{-1, +1\}^N$). The partition function normalizes the distribution so it is a distribution, and it is often considered as the holy grail of statistical physics because of all of the properties that one can calculate with it, such as free energy, entropy, and internal energy. However, it is hard to compute, as it is a sum over a whopping 2^N terms. Nevertheless, it turns out that in the 1D and 2D cases the partition function is analytically solvable, though the 2D case is a lot more challenging than the 1D one. For the 2D case and above, it is possible to use a mean-field approximation in which a lattice site is considered to experience the average behavior of its neighbors. While this approximation is qualitatively correct, it is only quantitatively correct asymptotically as the number of dimensions goes to infinity.

While phase transitions are some of the main phenomena that make the Ising model interesting, they are not the reason we are reviewing them here. Already, we can see some parallels between interaction energies and residue proximity. In a more general case of the Ising model where interaction terms $\boldsymbol{J}_{ij}$ can be unique to each pair of lattice sites i, j, one can tune these interaction terms in a way that favors a particular energy-lowering configuration of spins. There are a few steps needed to reach a more complete analogy, and Hopfield networks should be a helpful next step.

Hopfield networks, also known as the Ising model of a neural network, are fully connected undirected networks (e.g. every node has an edge connecting it to every other node). Historically, they have been developed to model neuronal firing during the retrieval of context-addressable memory, which is process of remembering a whole object after seeing only a part of it. Notwithstanding, they are a useful introduction to energy-based models.

Just like Ising models, Hopfield networks generate binary sequences $\boldsymbol{\sigma} \in \{-1, +1\}^N$. Unlike Ising models, which might only have nearest-neighbor interactions, Hopfield networks are fully connected and therefore account for 'long-range interactions'. More specifically, Hopfield networks are modeled after spin glasses, which themselves are a version of Ising models. Even more specifically, Hopfield networks follow the densely connected Sherrington-Kirkpatrick model for spin glasses as opposed to the Edwards-Anderson one (which only accounts for for nearest neighbors). Their energy function is the following:

$$E(\boldsymbol{\sigma}) = -\sum_{1 \leq i \leq j \leq N} J_{i,j}\sigma_i\sigma_j - \sum_i h_i\sigma_i. \tag{2.18}$$

Where the first term is the the sum of the interaction energies and the second is the sum of the external fields (also called biases). Once an input is given to the Hopfield network, it will update each of the nodes individually such that the energy is minimized until the system reaches a local minimum in its energy landscape (convergence guaranteed). Hebbian learning can be used to 'store' patterns of interest in the network by turning these patterns of interest into global minima in the energy landscape. Full binary images can be reconstructed when starting with information from only one half of these images. What if these stored patterns encoded protein sequences?

To be able to do so, we will need a Potts model. Potts models are generalizations of Ising models in which the sequences generated can be more than just binary $\boldsymbol{\sigma} \in \{1, \dots, A\}^N$. In our context, $A = 20$ for the 20 amino acids. In this new case, we write the energy function as

$$E(\boldsymbol{\sigma}) = -\sum_{1 \leq i \leq j \leq N} J_{i,j}(\sigma_i, \sigma_j) - \sum_i h_i(\sigma_i). \tag{2.19}$$

Notice here that $\boldsymbol{J}_{ij}(\sigma_i, \sigma_j)$ and $h_i(\sigma_i)$ are functions of both the positions and the characters occurring in these positions (i.e. the amino acids). One can think of $\boldsymbol{J}$ as a $N \times N$ matrix of $A \times A$ $\boldsymbol{J}_{ij}$ matrices. With this energy, we can write the probability of finding a sequence $\boldsymbol{\sigma}$ from a Boltzmann distribution as:

$$P(\boldsymbol{\sigma}) = \frac{1}{Z} \exp\left\{ \sum_{1 \leq i \leq j \leq N} J_{i,j}(\sigma_i, \sigma_j) + \sum_i h_i(\sigma_i) \right\} \tag{2.20}$$

This probability distribution is the crux of the EC framework. We started with a simple Ising model, added structure to it, and generalized it so that it matches our model of proximity constraints and evolution. This approach to predict residue contacts, protein folds, and the pathogenicity of variants is based on two conjectures: the first is that evolutionary processes *sample* sequences from a probability distribution defined by an energy landscape, and the second is that this energy landscape is defined by pairwise residue interdependencies as well as position-specific residue biases.

These conjectures might seem to be reductionist, but as the experimental results show, they seem to be reasonable approximations for the evolutionary machineries that generate protein sequences. Now that we have a model and it's probability distribution, the goal is to adjust the parameters of that model so that it fits the empirical observations: the frequencies and correlations seen in large homologous sequence alignments.

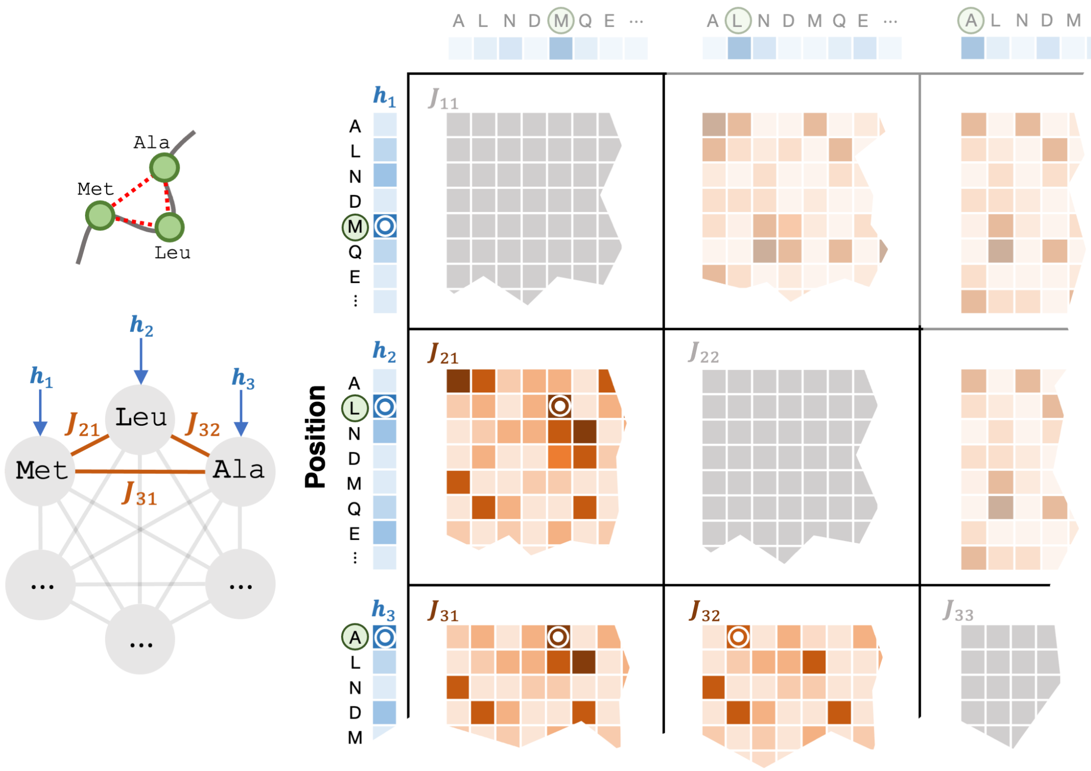


FIGURE 2.9: **Direct coupling analysis with a Potts model.** This is an example of DCA for the sub-sequence Met-Leu-Ala (MLA). Each position i has an external field h_i and each pair of positions (i, j) has an interaction energies matrix $\boldsymbol{J}_{i,j}$.

2.5.2 Direct Coupling Analysis from Sequence Covariation

Direct coupling analysis (DCA) is a broad term that designates the collection of strategies used to learn the parameters of the Potts model's distribution and predict structural properties from these parameters. Reiterating a major assumption of these strategies: evolution samples sequences from sequence space according to the energy-based probability distribution in eq. (2.20). Simply put,

$$\boldsymbol{\sigma} \sim P_{\boldsymbol{J},\boldsymbol{h}}(\boldsymbol{\sigma}). \tag{2.21}$$

The subscripts emphasize that the probability distribution $P(\cdot)$ is parameterized by the interaction terms $\boldsymbol{J}$ and biases $\boldsymbol{h}$, which are the ones we are trying optimize. For optimization, we will use an multiple sequence alignment $\{\boldsymbol{\sigma}^{(d)}\}_{d=1}^{D}$ with D sequences ('D' for depth) that are L residue-long each. Because we will need them later on, we define position-specific amino acid individual and pairwise frequencies and empirical correlation matrix as:

$$f_i(a) = \frac{1}{D}\sum_{d=1}^{D} \mathbb{1}_{\sigma_i^{(d)}=a} \tag{2.22}$$

$$f_{ij}(a,b) = \frac{1}{D}\sum_{d=1}^{D} \mathbb{1}_{\sigma_i^{(d)}=a}\mathbb{1}_{\sigma_j^{(d)}=b} \tag{2.23}$$

$$C_{ij}(a,b) = f_{ij}(a,b) - f_i(a)f_j(b) \tag{2.24}$$

One of the earliest DCA methods was mutual information (Marks et al., 2011), which is an information theoretic measure of the dependence between two random variables. In other terms, it quantitates the amount of information that we gain about one variable after observing another. Suppose $X \sim P_X, Y \sim P_Y$, and $(X,Y) \sim P_{X,Y}$, the mutual information of X and Y is:

$$\mathrm{MI}(X;Y) = \mathrm{KL}(P_{X,Y} \parallel P_X \otimes P_Y) \tag{2.25}$$

where $\mathrm{KL}(p \parallel q)$ is the Kullback-Leibler divergence between distributions p and q. For discrete distributions over a set of protein sequences and between positions i and j, we estimate the mutual information with

$$\mathrm{MI}_{ij}(X;Y) = \sum_{(a,b)\in\{1,\ldots,K\}^2} f_{ij}(a,b)\log\left(\frac{f_{ij}(a,b)}{f_i(a)f_j(b)}\right). \tag{2.26}$$

While mutual information seems like a reasonable measure to use for prediction, experimental results show that it is not. The reasons for its poor performance are manifold. First, mutual information does not rely on the Boltzmann distribution ascertained in the previous section. Contrary to approaches that exploit that distribution, mutual information is local, as it only relies on frequency estimates without fitting parameters that affect the likelihood of the entire sequence. Second, using correlation for constraint prediction is an inherently challenging problem. Transitive relationships between more than two positions significantly reduce the accuracy by increasing the false positive rate. Other problems such as noise, scarce data, and sampling bias also undermine the predictive power of DCA methods more broadly.

EVfold is one of the earliest methods developed to infer evolutionary couplings using the Potts model in eq. (2.20) (Marks et al., 2011). To do this, EVfold uses a new 'direct information' score based on the sequence probability based on the Potts model:

$$\mathrm{DI}_{ij} = \sum_{(a,b)\in\{1,\ldots,K\}^2} P_{ij}^{\mathrm{DI}}(a,b)\log\left(\frac{P_{ij}^{\mathrm{DI}}(a,b)}{f_i(a)f_j(b)}\right) \tag{2.27}$$

Where $P_{ij}^{\mathrm{DI}}(a,b)$ is introduced as an effective two-residue model with parameters inferred from the frequencies and correlations. Because computing the partition function Z is hard, optimizing $\{\boldsymbol{J},\boldsymbol{h}\}$ cannot be done via traditional likelihood maximization. Instead, the parameters of the distribution are inferred such that the distribution coincides with the individual and pairwise frequencies observed:

$$P(\sigma_i = a) = \sum_{\substack{\boldsymbol{\sigma}\in[K]^L \\ \sigma_i=a}} P(\boldsymbol{\sigma}) = f_i(a) \tag{2.28}$$

$$P(\sigma_i = a, \sigma_j = b) = \sum_{\substack{\boldsymbol{\sigma}\in[K]^L \\ \sigma_i=a \\ \sigma_j=b}} P(\boldsymbol{\sigma}) = f_{ij}(a,b) \tag{2.29}$$

Marks et al. use an mean-field approximation of the Potts model, which makes optimization quite fast. Under this approximation, the optimal $\hat{\boldsymbol{J}}_{ij}(a,b) = -(C^{-1})_{ij}(a,b)$ and $\hat{h}$ is then easily computed using the conditions in eq. (2.28). This two-residue effective model becomes:

$$P_{ij}^{\mathrm{DI}}(a,b) = \frac{1}{Z}\exp\left\{\hat{\boldsymbol{J}}_{ij}(a,b) + \hat{h}_i(a) + \hat{h}_j(b)\right\} \tag{2.30}$$

Ekeberg et al. (2013) employ a different approach to optimize for $\{\boldsymbol{J},\boldsymbol{h}\}$ which happened to outperform the mean-field approximation of Marks et al.: *pseudolikelihood maximization* (PLM). Maximum likelihood estimation requires us to maximize a likelihood that depends on the partition function (which is a function of $\boldsymbol{J}$ and $\boldsymbol{h}$, which we want to optimize). Instead of running into unwieldy sums, PLM maximizes the conditional probability of observing each amino σ_l acid given that the rest of the sequence $\boldsymbol{\sigma}_{\setminus l}$ is observed:

$$P(\sigma_l = \sigma_l^{(d)} \mid \boldsymbol{\sigma}_{\setminus l} = \boldsymbol{\sigma}_{\setminus l}^{(d)}) = \frac{\exp\left\{h_l(\sigma_l^{(d)}) + \sum\limits_{i\in[L]\setminus l} J_{il}(\sigma_i^{(d)}, \sigma_l^{(d)})\right\}}{\sum\limits_{a\in[K]}\exp\left\{h_l(a) + \sum\limits_{i\in[L]\setminus l} J_{il}(\sigma_i^{(d)}, a)\right\}}. \tag{2.31}$$

This conditional probability maximization is applied to every sequence in the MSA and to every residue within each sequence, yielding the pseudo log-likelihood:

$$\mathcal{L}_{\text{pseudo}}(\boldsymbol{J}, \boldsymbol{h}) = \sum_{d \in [D]} \sum_{l \in [L]} \log \left(P(\sigma_l = \sigma_l^{(d)} \mid \boldsymbol{\sigma}_{\setminus l} = \boldsymbol{\sigma}_{\setminus l}^{(d)}) \right) \tag{2.32}$$

Once the optimization is complete, a score S_{ij} is calculated for each pair ij by taking the Frobenius norm of the corresponding $\hat{\boldsymbol{J}}_{ij}$ matrix:

$$S_{ij} = \left\| \hat{\boldsymbol{J}}_{ij} \right\|_F = \sqrt{\sum_{a,b \in [K]} \hat{\boldsymbol{J}}_{ij}(a,b)^2}. \tag{2.33}$$

While there is a lot going on (and even more of it that is not shown), but the rationale is quite straightforward. Starting with the energy-based evolution assumptions, we try to find the parameters of the energy-based Potts model that explains the MSA best. Based on the approximated parameters and the resulting distribution, direct information or Frobenius norm scores are calculated, ranked, and the ones that are significantly large are considered predictive of residue contacts in the protein. Residue contact predictions are then translated into proximity constraints that are then used by folding algorithms to predict an accurate fold for a protein.

This method has also been used to predict residue contacts at the interfaces within protein complexes (Ovchinnikov et al., 2014; Hopf et al., 2014). The main modifications are in the data: to account for cross-protein interactions, the MSA is made of concatenated sequences observed in the same individual. The interaction matrices are then much larger, and are composed of intra-protein and inter-protein residue interactions.

Since residue-contacts within a protein or a protein complex are essential and conserved features of a coding sequence, they have been used to predict the effects of missense mutations in both single proteins (EVmutation) (Hopf et al., 2017), and protein complexes (EVcomplex) (Hopf et al., 2014). In short, the same energy-based model is fitted to the observed MSA of protein (complex) sequences, and the deleteriousness of a variant is estimated using a log-odds score of the variant and wild-type sequences, which is simply the difference in energy between the two sequences:

$$\Delta E(\boldsymbol{\sigma}^{\text{var}}, \boldsymbol{\sigma}^{\text{wt}}) = \sum_{1 \leq i \leq j \leq L} \left(\hat{\boldsymbol{J}}_{ij}(\sigma_i^{\text{var}}, \sigma_j^{\text{var}}) - \hat{\boldsymbol{J}}_{ij}(\sigma_i^{\text{wt}}, \sigma_j^{\text{wt}}) \right) + \sum_{i \in [L]} \left(\hat{\boldsymbol{h}}_i(\sigma_i^{\text{var}}) - \hat{\boldsymbol{h}}_i(\sigma_i^{\text{wt}}) \right) \tag{2.34}$$

EVmutation consistently outperformed SIFT, Polyphen-2, BLOSUM62, and a interaction energy-less energy-based model. The comparisons were made based on an array of mutagenesis experiments, in which proteins were mutated and their functions measured. Performance was assessed by taking the Spearman correlation between the predictive score and the measured function of the mutated proteins.

Notice how the log-odds score above not only depends on the variant, but on the entire sequence of the variant gene. This is an exciting shift to first-principled and model-based methods that are not only better-performing, but also remarkably more interpretable. However, a few issues remain. The first one is shared with most of the methods reviewed in this thesis; deleterious does not mean pathogenic, and pathogenic does not mean deleterious. Whiel EVcomplex and EVmutations can be used to predict the effects of *in vitro* protein mutations, their use in the clinic is not so straightforward.

Additionally, these methods cannot account for insertion-deletions that are not in the MSA because their underlying model is not *translationally invariant*, meaning that small-scale shifting of the inputs around the model's nodes (while preserving their relative order) will not yield the same results. The simple Ising model we considered first in eq. (2.16) *is* translationally invariant because all the interaction energies are the same across different pairs and the external field applied is uniform across the different lattice cites. The Potts model described in eq. (2.20), neither the interaction terms nor the external fields necessarily have any symmetry that would lead to translational invariance. All the methods reviewed in this thesis suffer from this drawback.

Another issue is that the Potts model on these energy-based models are based only accounts for pairwise interactions, while ignoring higher-order interactions. These higher-order terms are effectively impossible to estimate because they are hidden, and their number grow exponentially with the number of observed variables. In the the upcoming sections, we will review ways to approximate those hidden distributions in a way that is useful for prediction.

2.6 Deep Generative Models for Protein Evolution

Incorporating pairwise coevolution in algorithms for protein folding and pathogenicity prediction has proven to be greatly beneficial across the board. These energy-based models are grounded in the chemical and physical biology of the systems they deal with, and their increasing complexity mimics better and better that of the generative models we think underlie evolutionary sequence landscapes. But can we do better?

Epistasis is a broad term that describes the ways in which elements in a sequence interact with each other and affect each other's behavior, sometimes leading to very surprising outcomes in terms of fitness or function. These elements could be single nucleotides, codons, protein domains, regulatory elements, or any other genetic entity. Pairwise interactions are the simplest kind of epistasis. As we have seen in the previous section, amino acids that

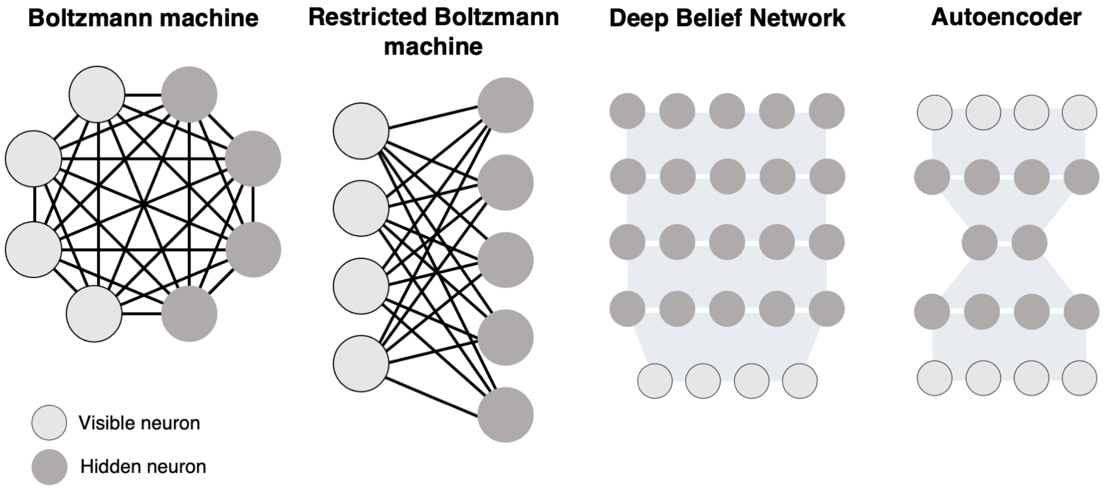


FIGURE 2.10: **A few (deep) generative models.** A Boltzmann machine (BMs) can be thought of as a stochastic Hopfield network with hidden nodes. Restricted Boltzmann machines (RBMs) are a more training-efficient and simpler version of BMs. A deep belief network is a stack of RBMs. A variational autoencoder (VAE) is based on variational approximation used to infer latent variables from an input, it is based on the autoencoder architecture in which an encoder compresses an input and a decoder reconstructs the input form the compression.

interact with each other in a protein fold tend to vary together to maintain or improve the structural and functional integrity of that protein fold. The same can be done with RNA molecules; sequence covariation has been used to estimate pairwise constraints (Weinreb et al., 2016). Higher-order epistasis is an even broader term that describes non-pairwise coupling; i.e. when *groups* of positions interact with and modulate each other. It can also be thought of as the 'coevolution of coevolutionary mechanisms', and the patterns it leaves in the data could be thought of as 'correlations of correlations'. Several studies have demonstrated a substantial influence of higher-order epistasis on enzymes (Yang et al., 2019), transcription factor (Anderson et al., 2015), and RNA evolution (Bendixsen et al., 2017; Weinreich et al., 2013).

Higher-order epistasis is considerably harder than pairwise epistasis because it is further removed from the data, and it's effects are tougher to discern and disentangle from noise and spurious correlations. Additionally, the number of possible higher-order interactions grows astronomically large as the length of sequences increases, and given that these are all hidden variables, the problem of optimizing over them easily becomes intractable. Going beyond the observable and into the realm of hidden variables is a daunting task, and will have to rely heavily on reasonable approximations.

The purpose of this section is to review two recent deep generative models (DGMs) for protein evolution, phylogeny reconstruction, and variant pathogenicity prediction. But first, it would be helpful to see how we got from shallow to deep generative networks while introducing one of the most commonly used DGMs: the Variational Autoencoder (VAE).

2.6.1 From Boltzmann Machines to Variational Autoencoders

There is no obvious way to connect Boltzmann Machines to VAEs; the former is an energy-based generative model while the latter is a directed probabilistic graphical model whose posterior is approximated with variational inference and a neural network. Nevertheless, one can still draw a qualitative path from one to the other, bridging them through more similar intermediates.

Picking up from where we left off in Section **2.5.1**, our bridge starts at Hopfield networks. With a simple Hebbian learning rule, these energy-based deterministic models were able to memorize patterns and retrieve these patterns from incomplete or scrambled inputs. But there was one problem: training was never perfect and the derived energy landscapes would often have unwanted local minima that would trap the Hopfield network and prevent it from reaching the sought-after global minima.

The solution to this problem was simple and effective: the networks were made probabilistic instead of deterministic, meaning that sampling states was now a stochastic process in which the probability of a unit being flipped to -1 or $+1$ depended on the 'field' experienced at that unit. This Gibbs-like sampling would go on and on, ideally jumping out of local minima and reaching global ones. This improved network was given hidden units to increase its memory capacity, and the resulting network was called a Boltzmann machine.

Boltzmann machines (BMs) are fascinating because of the Hebbian nature of their training and their runtime behavior. However, they were really hard to train because that process required a positive and a negative phase which required letting the network run freely for every pattern in the training set. This drawback made them impractical for machine and deep learning purpose, but it also motivated the development of training-efficient restricted Boltzmann machines (RBMs).

Unlike Boltzmann machines, RBMs are not fully connected: their units (also called *neurons*) form a bipartite graph with the two groups being the visible and the hidden neurons, such that there are no connections within the same group. This bipartite graph setup is relatively easily trained using a gradient-based contrastive divergence algorithm and Gibbs sampling. RBMs can be trained in a supervised (with labels) and unsupervised manner, depending on the task at hand. Their depth can be increased by adding extra layers of hidden variables to carry out deep learning tasks, in which case RBMs can be stacked together to create deep belief networks (DBNs) which can

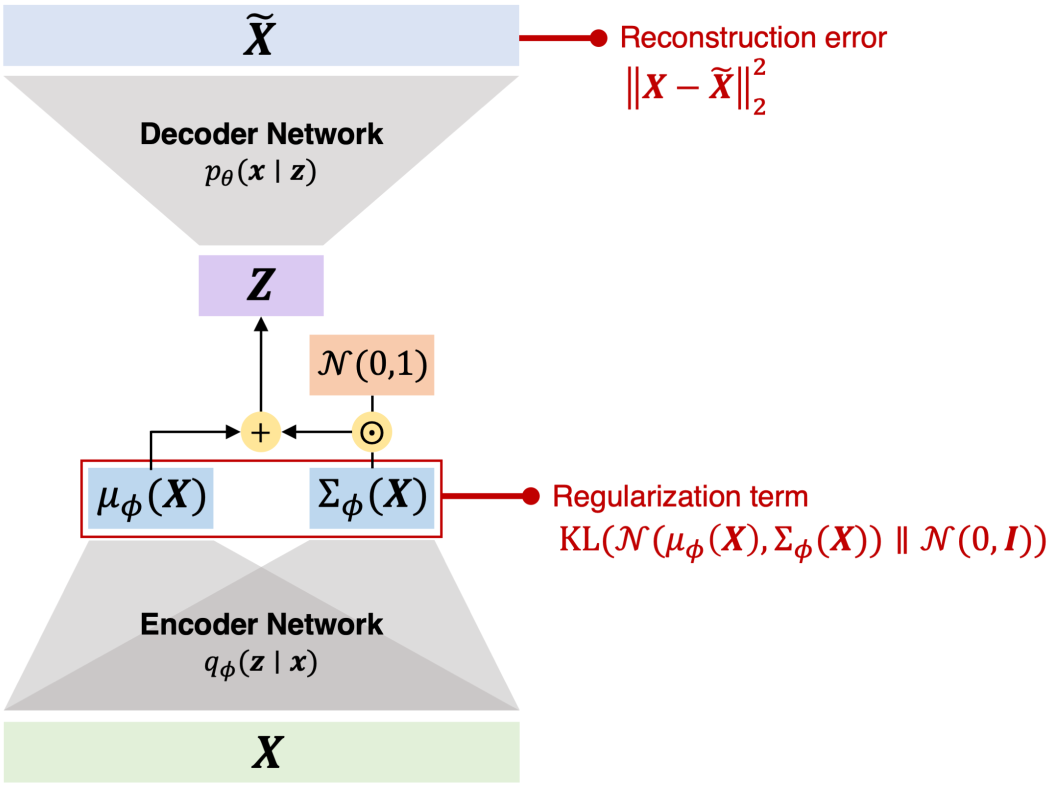


FIGURE 2.11: **The anatomy of a variational autoencoder.** Much like an autoencoder, a variational autoencoder consists of an encoder network that compresses the input (by finding the underlying latent variables) and a decoder network that reconstructs the input from the inferred latent variable. Unlike an autoencoder, a VAE samples the latent variables from a probability distribution (here depicted as a multivariate Gaussian), and the loss is essentially composed of the reconstruction loss and the KL divergence between the inferred distribution of the latent variables and their prior distribution (here, $\mathcal{N}(0, I)$).

in turn be greedily trained layer-wise.

DBNs are powerful deep generative models wherein each layer learns features from the previous one. However, DBN neurons are binary and model architecture can be unflexible. An alternative, real-valued, and poweful deep generative model that can accommodate various neural network architectures is the Variational Autoencoder.

2.6.2 Dissecting the VAE

As Kingma and Welling write in a recent introduction to the subject, VAEs marry graphical models and deep learning by using deep neural networks to perform variational inference on latent-variable graphical models. (Such a graphical model is shown in FIGURE.) In short, VAEs approximate the posterior distribution of latent variables (conditioning on the observed data); they optimize an encoder that meaningfully embeds data points $X \in \mathcal{X}$ into a latent space that the decoder understands and uses to reconstruct and generate data points. Such a model fits perfectly in the context of variant pathogenicity prediction because it has the ability to match the complexity of the true generative models that produce the sequences we observe. This brief review was written with ideas from Kingma and Welling (2019); Blei et al. (2017); Doersch (2021).

VAEs try to solve the problem of intractable posteriors in Bayesian graphical model. Consider the model in FIGURE. Conditioning on the data $\boldsymbol{X} = \boldsymbol{x}$, we would like to estimate the posterior distribution

$$p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z})}{p_\theta(\boldsymbol{x})}, \tag{2.35}$$

where the subscript θ emphasizes the parameters of the probability distribution, and $P(\boldsymbol{z})$ is the prior density on the latent variables. The numerator, which is the product between the prior and the likelihood, is easy to calculate, whereas the denominator can be quite trickier:

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}. \tag{2.36}$$

In fact, in most cases, the denominator is intractable because it considers all possible explanations (latent variable combinations) for $\boldsymbol{x}$, which can grow exponentially with the number of latent variables or their support. A concrete example would be one wherein $\boldsymbol{z} = (z_1, \ldots, z_k)$, which would yield a hulking mess

$$p_\theta(\boldsymbol{x}) = \int \left(\ldots \left(\int \left(\int p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z}) dz_1 \right) dz_2 \right) \ldots \right) dz_k. \tag{2.37}$$

Instead of (not) finding intractable solutions, we can use variational inference that seeks to approximate them. We do so by specifying a family of densities $\mathscr{D}$ that we can use to approximate the posterior distribution. The goal is to find the one that approximates the posterior 'best' by minimizing the Kullback-Leibler divergence between the approximation and the true posterior:

$$q^*_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = \underset{q_\phi \in \mathscr{D}}{\arg\min}\, \mathrm{KL}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p_\theta(\boldsymbol{z} \mid \boldsymbol{x})). \tag{2.38}$$

This divergence is, in turn, equal to

$$\mathrm{KL}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p_\theta(\boldsymbol{z} \mid \boldsymbol{x})) = \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{p_\theta(\boldsymbol{z} \mid \boldsymbol{x})}\right] \tag{2.39}$$

$$= \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log p_\theta(\boldsymbol{z}, \boldsymbol{x})\right] + \log p_\theta(\boldsymbol{x}). \tag{2.40}$$

This Kullback-Leibler divergence is intractable because it depends on $\log p_\theta(\boldsymbol{x})$. However, $\log p_\theta(\boldsymbol{x})$ is just a constant that depends on the data. Rearranging terms, we get

$$\log p_\theta(\boldsymbol{x}) = \mathrm{KL}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p_\theta(\boldsymbol{z} \mid \boldsymbol{x})) + (\mathbb{E}_{\boldsymbol{z}\sim q}\left[\log p_\theta(\boldsymbol{z}, \boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\right]) \tag{2.41}$$

$$\geq \underbrace{\mathbb{E}_{\boldsymbol{z}\sim q}\left[\log p_\theta(\boldsymbol{z}, \boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\right]}_{\mathrm{ELBO}(q_\phi)} \tag{2.42}$$

Because Kullback-Leibler divergence is always positive, the remaining term is called the evidence lower bound (ELBO) of q_ϕ because it is a lower bound for $\log p_\theta(\boldsymbol{x})$ and $p_\theta(\boldsymbol{x})$ is called the 'evidence' in eq. (2.35). The ELBO has several compelling properties. First, maximizing it will allow us to get a better estimate of the evidence (because it is a lower bound for it). Second, because the LHS is a constant, maximizing the ELBO is equivalent to minimizing the Kullback-Leibler divergence between the true posterior and our approximation of it. Using the definition of joint probability to decompose the first term, we can rewrite the ELBO as:

$$\mathrm{ELBO}(q_\phi) = \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right] + \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log p_\theta(\boldsymbol{z})\right] - \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log q_\phi(\boldsymbol{z} \mid \boldsymbol{x})\right] \tag{2.43}$$

$$= \mathbb{E}_{\boldsymbol{z}\sim q}\left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right] - \mathrm{KL}(q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \parallel p_\theta(\boldsymbol{z})). \tag{2.44}$$

Such a decomposition of the ELBO exposes two terms: the expectation of the log-likelihood and the Kullback-Leibler divergence between the latent variable prior density and our approximation of the latent variable posterior density. Therefore, maximizing the ELBO entails maximizing the expected log-likelihood and minimizing the divergence between the prior and approximate posterior, which intuitively is what we want to do.

All the above is the 'variational' aspect of a VAE; the 'autoencoder' aspect arises when decide to parameterize the distributions p_θ and q_ϕ as neural networks. Typically, we choose our *inference model* q_ϕ such that

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \mu_\phi(\boldsymbol{x}), \Sigma_\phi(\boldsymbol{x})), \tag{2.45}$$

Where $\mu_\phi(\cdot)$ and $\Sigma_\phi(\cdot)$ are neural networks that take an input $\boldsymbol{x}$ and return a vector of means and variances. And we choose our *generative model* p_θ to be a neural network as well. Usually, the prior density over $\boldsymbol{z}$ is a multivariate normal with mean $\mathbf{0}$ and covariance matrix set as the identity matrix.

The high complexity of VAEs, their layered nature, and their capacity to scale up make it a very compelling model for higher-order epistasis. Each layer learns new hidden features starting from the previous layer, and this hierarchy of latent factors influence sequences in a way similar to that of higher-order epistasis.

2.6.3 Evidence Lower Bounds to Predict Variant Deleteriousness

Even though we are now equipped with a powerful computational technique to account for latent variables, the paradigm for pathogenicity prediction hasn't changed: it is still based on the assumptions that evolution is a generative process that samples sequences from sequence space according to a probability distribution over that space, and that this probability distribution is proportional to the fitness conferred by these sequences. The VAE framework is but a way to estimate this probability distribution while accounting for higher-order epistasis and other hidden variables, thereby better mimicking the generative processes underlying the data.

From the assumptions stated above, we can expect pathogenic (fitness-lowering) sequences to have a lower sampling probability than fitness-neutral or fitness-improving sequences. In their method, DeepSequence, Riesselman et al. (2018) formalize this heuristic by considering the log-ratio

$$\log \frac{p_\theta(\boldsymbol{x}^{\text{var}})}{p_\theta(\boldsymbol{x}^{\text{wt}})}, \tag{2.46}$$

which should be negative for unfavorable variants $\boldsymbol{x}^{\text{var}}$ and non-negative otherwise. This log-ratio is identical to the one in eq. (2.34), which in the pairwise Potts model case was a difference in energies (Hopf et al., 2017). In this case, the probabilities in that log-ratio are replace by the ELBO itself, as it is a lower bound for $\log p_\theta(\boldsymbol{x})$.

Deep learning is often described as a black box, for what goes on inside of it is rarely human-interpretable. To counter this, Riesselman et al. added to their model an assortment of priors and parameterizations to their model:

- Group sparsity priors: Based on the assumption that small groups of latent variables influence only a few positions at a time. This prior is imposed on the last layer of the decoder neural network for $p_\theta(\boldsymbol{x})$.

- Encouraging correlations in amino acid usage: At each position within a protein sequence, the matrix that transforms the hidden layer's activations into a distribution over the amino acids at this position is parameterized as the matrix product of a global 'dictionary' $\boldsymbol{C}$ (global in that it is used for all positions) and a position-specific matrix of weights. This global dictionary of size $20 \times E$ (where 20 is the number of possible amino acids and E is an arbitrary hyper-parameter) can be interpreted as a table of global features for each amino acid. After training, amino acids known to have similar properties had similar features in this global dictionary.

- A lot of priors: Normal priors were placed over the values in global dictionary, the final layer weights, and the sparsity parameters 'gating' said weights. Setting such priors added a term in the ELBO that represented the divergence between the inferred parameters and their prior distribution.

Unsurprisingly, DeepSequence outperforms both pairwise (EVmutation) and position independent mathods (SIFT, Polyphen-2, BLOSUM62) on the vast majority of data sets drawn from various controlled mutagenesis experiments. Performance was measured through the correlation between the log-ratio score and the experimental fitness loss. Interestingly, DeepSequence was less accurate with viral proteins than other methods on some experiments. Additionally, a feature ablation experiment was carried out, in which the model was trained and tested with different combinations of the priors and parameterizations outlined above. Results showed that all three of them added accuracy to the model while making it more interpretable at the same time.

Riesselman et al. also visually inspected the latent space of β-lactamase sequences from many organisms by using only 2 hidden features (rather than 30 in the actual model). As expected, the latent variables were concentrated in a tight region around their prior density, but the visualization also revealed an organization of sequence clusters that were highly consistent with phylogenetic knowledge. Sequences from related similar bacterial species were clustered right next to each other. The star-shaped latent variable point cloud was also inspiring; could one recover refined phylogenetic structure from the spikes diverging from the center of that star-shaped point cloud?

2.6.4 Latent Space, Fitness Landscapes, and Phylogeny

Embedding roughly refers to the process of mapping vectors from a high dimensional space to a low dimensional one while preserving the structure found in high dimensions. VAEs are powerful inference and embedding tools, as the latent variables they infer can be used to embed long, discrete sequences in a real-valued latent space (the vector space in which the latent variables $\boldsymbol{z}$ can be found).

Using simulated and real sequences, it was shown that training a VAE model on large protein sequence data resulted in a structured latent space that reflected the evolutionary phylogenetic relationships between the sequences in that data (Ding et al., 2019). This model was called protein evolution VAE (PEVAE). Visualizing this latent space revealed star-shaped point clouds that were ordered in a fashion consistent with the generative phylogenetic models that produced these point clouds in the first place. It was also observed that the points close to the center of these star-shaped distributions tended to be constituted of mostly root node sequences, indicating that ancestral relationships between sequences can be recovered in latent space. Several features from simulated and realistic phylogenetic trees were recovered in the embedding. While the resolution of these features could have been higher, the preliminary results are very promising.

Ding et al. also made use of a semi-supervised learning framework to model the fitness landscape of the sequences. The VAE was first trained using unlabeled sequences to learn a meaningful embedding latent space, an encoder, and a decoder. Then, a Gaussian process was fitted to variant protein melting temperatures gathered from controlled mutagenesis experiments and embedded in the latent space via the encoder. The results were equally as promising, as the predicted temperatures correlated well with the actual temperatures.

In the same study, the protein stability of fibronectin type III and staphylococcal nuclease variants was estimated using the marginal probability of those sequences $p_\theta(\boldsymbol{x})$ given by the VAE. This marginal probability was estimated using importance sampling with a proposal distribution $\boldsymbol{z} \sim q_\phi(\boldsymbol{x} \mid \boldsymbol{x})$:

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} d\boldsymbol{z} \tag{2.47}$$

$$\approx \frac{1}{S} \sum_{i=1}^{S} \left[\frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \right] = \frac{1}{S} \sum_{i=1}^{S} \left[\frac{p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z})}{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})} \right] \tag{2.48}$$

The Pearson's correlation coefficients were calculated based on the change in probability and change in experimentally measured free energies, resulting in a correlation of 0.81 for fibronectin type III and 0.50 for the staphylococcal nuclease. These encouraging results confirm that while protein stability is an important driver for evolution, it is not the only one.

2.7 A Note on Sequence Weighting

Like with any statistical learning method, the performance of homology and conservation-based tools is highly dependent on the quality of the multiple sequence alignments used for training and parameter-tuning. As we saw in

Section 2.1, the sequences used should be independently sampled and identically distributed (the i.i.d. assumption) to be representative of their real-world distribution $\mathcal{D}$. However, this assumption is rarely reasonable with the current data sets: much effort is still needed to reach a level of sequence diversity that reflects the diversity of human populations and significant amounts of phylogenetic bias might be present in the data. This is seen especially in databases of sequences from related species which did not have enough time for independent evolution.

To account for this source of bias, many methods employ a sequence weighting scheme that generally upweights uncommon sequences and downweights common ones. There are two main weighting schemes used by the generative methods discussed in Section 2.5 and Section 2.6. The first one is used by mean-field DCA (Marks et al., 2011; Morcos et al., 2011), pseudolikelihood maximization DCA (Ekeberg et al., 2013; Hopf et al., 2017), and DeepSequence (Riesselman et al., 2018). Simply, each sequence $\boldsymbol{\sigma}^{(d)}$ in the MSA $\{\boldsymbol{\sigma}^{(d)}\}_{d=1}^{D}$ is weighted inverse-proportionally to the number of 'similar' sequences in the MSA, where two sequences are 'similar' if their percentage identity is higher than some percentage. Alternatively, two sequences are 'similar' if they have a Hamming distance D_H below some threshold τ. The resulting weights are therefore

$$w^{(d)} = \left(\sum_i \mathbb{1}_{D_H(\boldsymbol{\sigma}^{(d)}, \boldsymbol{\sigma}^{(i)}) < \tau} \right)^{-1}, \tag{2.49}$$

and the effective sample size is defined as

$$N_{\text{eff}} = \sum_{d=1}^{D} w^{(d)}. \tag{2.50}$$

How these weights are used depends on the method. Marks et al. and Morcos et al., whose methods rely on the observed position-specific and pairwise frequencies, uses the weights and the effective sample size to adjust the these observed frequencies to perform mean-field approximation DCA. Ekeberg et al. and Hopf et al. use those frequencies to weight the sequence log-likelihoods in pseudolikelihood maximization DCA. In DeepSequence, Riesselman et al. turn these weights into probabilities $p^{(d)} = w^{(d)}/N_{\text{eff}}$ and use these probabilities to sample minibatches during training.

The second sequence weighting scheme is used in PEVAE (Ding et al., 2019), and is based on Henikoff and Henikoff (1994). The scheme works as follows, each position $\sigma_j^{(d)}$ in the MSA is weighted by a weight $w_j^{(d)}$, such that

$$w_j^{(d)} = \frac{1}{C_j C_j^{(d)}}, \tag{2.51}$$

where C_j is the number of unique amino acids observed at position j in the MSA and $C_j^{(d)}$ is the number of sequences in the MSA that have the same amino acid as $\sigma_j^{(d)}$ at the same position. The weight of a sequence is therefore the sum of the position-wise weights normalized across all the sequences

$$w^{(d)} = \sum_{j=1}^{L} w_j^{(d)} \Big/ \sum_{d=1}^{D} \sum_{j=1}^{L} w_j^{(d)}. \tag{2.52}$$

These weights were incorporated in the ELBO objective for training. Experimental results show that all these weighting schemes help with generalizability, but whether they are optimal is not obvious. By downweighting common sequences and upweighting less common ones, these schemes might to assume 'flatter' distributions of sequences in the real world, which might introduce some biases elsewhere.

This extensive review has allowed us to identify a common roadmap, or framework, to understand variant pathogenicity methods in a broader and more general context of analysis. This context of analysis includes not only performance, but also the sample complexity of a model, the type(s) of data or features used, and the assumptions about this data. We will keep these insights in mind throughout the upcoming chapters, where I reproduce and examine in-depth a classification method for HCM-related variants, and introduce novel approaches for deep learning models in clinical genomics.

3

Establishing the Baseline Model: Reproducing PolyPhen-HCM

PolyPhen-2 predicts deleteriousness, but that may not be enough to predict the pathogenicity of a variant, mainly because functional deleteriousness and pathogenicity are not equivalent (Eilbeck et al., 2017). To demonstrate how such a method could be used for variant interpretation, it was integrated into another machine learning algorithm to interpret variants found in six genes that have been strongly associated with hypertrophic cardiomyopathy (HCM), a relatively common inherited structural heart disease (Jordan et al., 2011). These six genes are all part of the cardiac sarcomere and contractile apparatus, including the myosin light and heavy chains, troponins, a tropomyosin, and a myosin binding protein.

This algorithm, PolyPhen-HCM, is one of the earliest examples of its kind, as it attempts to predict the clinical significance of HCM-specific variants based on 4 features: (1) the PolyPhen-2 score at the relevant position, (2) a structure score calculated using the inferred conformational changes of select PDB folds, (3) a coiled-coil tendency-based score, and (4) sitewise evolutionary substitution rates computed using a Markov Chain Monte Carlo (MCMC) algorithm in the MrBayes software package. The discriminative model in PolyPhen-HCM is a Support Vector Machine (SVM) classifier which predicts whether a variant is pathogenic or not based on those 4 features. SVMs are traditionally linear; the most basic decision boundary they can learn is a linear hyperplane (a line in 2 dimensions, a plane in 3 dimensions). To increase the complexity of their hypothesis class, one can map the existing features into a higher-dimensional space in which learning and prediction happen. This can be done by creating new features as functions of other features. Thankfully, the 'kernel trick' allows us to learn in the higher-dimensional space without the need for such mapping. Intuitively, a kernel is a function that calculates some definition of distance (formally, the dot product) between two points. The SVM in PolyPhen-HCM uses the radial basis function (RBF) as a kernel (as opposed to a linear or polynomial kernel; see **??**). Intuitively, the RBF kernel-SVM behaves like a nearest-neighbor classifier, where data points that occur in the vicinity of a certain category are predicted to be of said category. The resulting classifier predicts each variant to be either benign or pathogenic.

The confidence of a prediction for a data point depends on its position relative to the decision boundary in the feature space; the farther a variant is from this boundary, the more confident our prediction would be. To increase accuracy and to account for low-confidence predictions, Jordan et al. added a third class, 'no call', in which the classifier rejects the variant. Of course, such a strategy variants will decrease the coverage of the classifier, since variants that fall in the no call zone will be rejected, and fewer variants overall will be predicted as benign or pathogenic.(See Figure **3.4** for a simple visualization of coverage and accuracy.)

The results were promising: the classifier was only able to reach an accuracy of 92% when the coverage was 57%. This means that 43% of the variants were not given a prediction. At 100% coverage (no middle ground; all variants are classified as either pathogenic or benign), the accuracy was around 77%, which is too low for the safe clinical use of the method (see FIGURE for an explanation of how coverage and accuracy change with each other). These shortcomings can be attributed to a multitude of factors. First, the 'ground-truth' labels were manually determined using a simplistic flow chart that takes into account population frequencies, segregation patterns, and functional evidence. However, as we saw earlier, such data can be problematic for many reasons: population frequencies were significantly less accurate before the advent of large and diverse genome aggregation databases such ExAC and gnomAD, and our knowledge of the mechanisms underlying most Mendelian diseases remains quite poor. In addition, it is the unreliability of such simplistic flowcharts that has motivated the development of computational methods of prediction, so relying on those same flowcharts to evaluate the performance of our algorithms could be detrimental to computational learning.

Moreover, the training data was limited, consisting of only 74 variants from 6 genes. It was also unbalanced; only 7/74 variants were labeled as benign, whilst 41/74 were labeled as pathogenic, and the remaining 26/74 were labeled as likely pathogenic. The representation of the 6 HCM genes was also somewhat unbalanced, with 22/74 variants coming from 4 genes and the remaining 52 coming from only 2. (Nevertheless, this distribution roughly correlates with the length of each gene, so this imbalance might be innocuous.) The hypothesis class of the SVM classifier could have also been inappropriate for the task at hand. However, the complexity of that classifier was suitable for the small-sized training data; a more complex model could have easily overfitted.

Another issue that might be more pertinent to this thesis is the fact that PolyPhen-HCM, just like many other classifiers of its kind, does not take epistasis into account. The algorithm takes a single-nucleotide variant as input and outputs a score, based on which we can classify this variant. However, genetic tests are usually done on an array of genes rather than a single one and many variants can be found in an individual. Without knowledge about how these variants interact with each other, our predictions are missing crucial information about genomic context.

Despite the limitations above, PolyPhen-HCM performed reasonably well. Its approximation of sequence evolution as a position-specific and independent process is still powerful enough to reach an intermediate level of performance. The purpose of this chapter is fourfold: the first is to reproduce the PolyPhen-HCM classifier while using recent variant data from NCBI ClinVar and gnomAD. I had to re-design most of the code, statistical methods, and algorithms used in PolyPhen-HCM starting from first principles. The code I wrote is now hosted on GitHub at `https://github.com/RalphEST/my-polyphen-hcm`, available for anyone to use, improve and learn from. The second purpose of this chapter is to delve deeper into the original results found in Jordan et al. (2011). More specifically, I examine a more comprehensive array of performance statistics to evaluate the secondary effects of increasing accuracy at the expense of coverage. The third purpose of this thesis is to test the same method with more recent variant data from the NCBI ClinVar and gnomAD variant databases. The fourth purpose is to establish a baseline performance for later comparisons and explore first-hand some of the practical questions and paradigms encountered in variant pathogenicity interpretation.

3.1 Methods

PolyPhen-HCM uses 4 position-specific features, namely the PolyPhen-2 score, the coiled-coils tendency score, the structure score, and an estimate of the rate of evolution of a position. Two of these features are variant- and position-specific, specifically the PolyPhen-2 and the coiled-coils tendency scores, meaning that not only the position of a variant matters, but the resulting variant sequence as well. The remaining two features are only position-specific, meaning that the amino acid change brought by a variant is not incorporated in the score.

To reproduce the algorithm, I created a collection of programs that starts with a list of variants annotated using HGVS (Human Genome Variation Society) notation, and successively computes the 4 features for each variant. For example, `NM_000257.4(MYH7):c.5774G>A(p.Arg1925His)` is the HGVS notation for a variant in the gene MYH7 (the cardiac myosin heavy chain 7) that substitutes nucleotide G at position 5774 in the cDNA with an A, which subsequently switches the arginine (Arg) at position 1925 in the translated protein to a histidine (His). Then, based on the variant features and known clinical significance (e.g. the labels that we would like to predict), I train a variety of SVM models and compare them against one another.

3.1.1 The (Labeled) Data

In addition to the 74 variants used in the original PolyPhen-HCM paper, we sought to add more data to improve balance and accuracy. To do so, we used variant data from ClinVar and gnomAD.

The data from the NCBI ClinVar database was collected following a comprehensive search for missense (protein coding) variants in genes associated with Hypertrophic Cardiomyopathy (HCM). Each variant has its own HGVS notation as well as other annotations that identify it. However, as it is the case with most data in bioinformatics and elsewhere, significant data wrangling was needed to ensure that the annotations are standardized and in conformity with the pipeline's expectations. Each variant is also labeled with a manually predicted pathogenicity classification, with classes including '(likely) benign', '(likely) pathogenic', and 'of unknown significance'. Variants with conflicting evidence are labeled as such. To compensate for the fact that ClinVar variants are also labeled manually by the laboratories that submit them, we refined our search by only selecting variants with at least 2/4 'stars'; that is, variants that have multiple submitters with no conflicts in interpretation. This refining in our search criteria produced a data set that is about half the size of the one we started with, which speaks to how inconsistent our knowledge of human genetic variation can be. Throughout our searches on ClinVar, we noticed that roughly half of the variants identified are of unknown significance and that a quarter of the remaining half have conflicting interpretations — leaving us with only a few hundred variants that are reliable enough for supervised model training.

Even when including material from ClinVar, benign variants constituted a small minority of our data. This might be because of *ascertainment bias*: ClinVar is a database of variants that might be linked to disease, it is therefore inherently biased toward pathogenic variants and against benign ones. Genetic testing laboratories primarily look for pathogenic variants, not benign ones, which are often the result of changing our interpretation of a pathogenic variants. To adjust this imbalance, we decided to use variants discovered in gnomAD. The majority of the polymorphism that are seen in this aggregation database had never been seen before, and very little is known about their clinical significance Karczewski et al. (2020). However, we managed to find variants that are likely to be benign based on their frequency of occurrence in the database, the prevalence of HCM in the population, and its penetrance (the probability of disease given the variant). Intuitively, the frequency of a causative variant is upper-bounded by the prevalence of HCM times the inverse of its penetrance; a variant whose frequency exceeds this upper bound is thus likely to be benign. Using the method and calculations from Whiffin et al. (2017), we decide to set the conservative threshold at 4×10^{-5}, meaning that any relevant gnomAD variant with a frequency higher than this threshold will be included in our analysis. This assumption allowed us to gain 252 variants, some of which were already in our ClinVar data.

How the training data is labeled is a key factor in the success (or lack thereof) of supervised training tasks, and even more so when it comes to variant classification not only because we know little about how variants interact with each other, but also because our labels can themselves be incorrect. This observation provides even more context to our discussion of the energy- and latent variable-based generative models described in Chapter **2**. Not only are they better approximations of evolution, they are also unsupervised, as their training does not require knowledge about a sequence's induced phenotype. Instead, they attempt to unveil a structure that is intrinsic to the data, such as pairwise covariation between residues in a protein (complex). Once this structure is unveiled, i.e. once we have a generative model (be it an energy-based distribution or a variational approximation of latent posteriors), we can model the target outcomes as a function of this underlying structure. For instance, DeepSequence and EVcouplings both model evolution as a fitness-maximizing sampler from a distribution they approximate, and subsequently use this distribution as a proxy for sequence fitness — and pathogenicity.

Lastly, the data in Jordan et al. (2011) is composed of variants from six genes. However, the original set contained two more (ACTC1 and MYL3) because of the extreme scarcity of their variants. Because our ClinVar data set is larger, these two genes will be included in our analysis. (Though they still contribute relatively few variants.)

3.1.2 The Features

PolyPhen-2 and other annotations

Instead of recomputing the PolyPhen-2 scores for each variant in our data, we collected precomputed PolyPhen-2, SIFT, BLOSUM62, and CADD scores from the Ensembl VEP (variant effect predictor). For every variant in HGVS notation, the VEP returns the set of gene transcripts it affects. Because genes sometimes overlap, a variant can affect many genes in different ways, and because a single gene can produce different transcripts (e.g. alternative splicing), a single variant can affect many transcripts in different ways. After a transcript is spliced and processed, it is usually referred to as an isoform to highlight the fact that a gene can have a multitude of isoforms. In our analysis, we only consider the transcripts and isoform that derive from the eight genes of interest.

Because PolyPhen-2 is variant-specific (as opposed to only position-specific), it might return different probabilities for each gene isoform, which might present an issue. However, the probabilistic nature of its Naïve Bayes classifier is helpful: PolyPhen-2 returns with its classification the probability that a variant is deleterious to the function of its protein isoform i, which we write as $P(\text{Del} \mid \text{Iso}_i)$, and we are interested in the probability that a protein will loose its function — considering all the possible isoforms, or $P(\text{Del})$. A straightforward way to get from the former to the latter is by using the law of total probability:

$$P(\text{Del}) = \sum_i P(\text{Del} \mid \text{Iso}_i) P(\text{Iso}_i), \tag{3.1}$$

where $P(\text{Iso}_i)$ can be interpreted as the abundance of isoform i relative to other isoforms of the same gene. Because there is not much data on isoform-specific expression levels, we will assume all isoforms are equally likely, which makes $P(\text{Del})$ a simple average of the PolyPhen-2 probabilities over all isoforms. Surprisingly, this neither happens with SIFT nor CADD, the latter of which is built on top of PolyPhen-2. BLOSUM62 returns the same score as long as the amino acid substitution is the same across transcripts, which is always the case.

MrBayes

MrBayes is a software package that uses Markov Chain Monte Carlo (MCMC) methods to optimize the parameters of an evolutionary model for a multiple sequence alignment. While most of the sequences in the MSA were manually curated in Jordan et al. (2011), we wanted to make the process fully automated. For each of the eight HCM-associated genes, homologous sequences were found using the NCBI BLASTP tool agains the NCBI database of non-redundant sequences and an MSA was constructed using the Clustal Omega tool from EMBL-EBI. To make sure the sequences were orthologous and with identical function, only sequences with over 80% coverage with the query sequence were selected. In addition, sequences from genes that are 10% longer or shorter than the query sequence were discarded. (This percentage was chosen arbitrarily, and there are better ways to ensure the functional similarity of sequences, such as by only keeping sequences with the same Pfam domains.) The resulting MSA contained around 230 sequences from a diverse set of taxons, with identities ranging from 84% to over 99%.

The MrBayes package supports a wide variety of evolutionary substitution models, ranging from the simplest Jukes-Cantor 1969 (JC69; Jukes and Cantor (1969)) model to the General Time-Reversible (GTR) model. The amino-acid model we chose is the same in Jordan et al. (2011); it is a modified version of the Felsenstein 1981 (F81; Felsenstein (1981)) model in which the stationary distribution of amino acids is allowed to be non-uniform and the overall rates vary per position according to a variable r_i. MrBayes estimates those rate variables using a Metropolis-coupled MCMC, and we use said rates a feature in our variant classifier.

The Coils Score

Four of the six proteins of interest, namely MYH7, TNNI3, TNNT2, and TPM1, have coiled-coil regions that are essential to their functions. A variant that interferes with the folding of these coiled-coil regions could have deleterious effects, and therefore be pathogenic. To account for such effects in their classifier, Jordan et al. use another software package called COILS (Lupas et al., 1991). COILS compares a sequence to a database of proteins that contain coiled-coils and computes a similarity score which in turn is compared to the distributions of scores from globular and coilded-coil proteins. Based on the latter comparison, COILS returns the probability that a position in the sequence will occur in a coiled coil.

PolyPhen-HCM uses the score that COILS returns (and not the probability). The feature used in the final classifier was the magnitude of the larges sing-residue change. That is, the scores are compared across the sites of two sequences that differ at the variant position (one with the reference amino acid, the other with the variant one), and the largest signed difference is used in prediction, which we will call the ‘maxdelta’ score. In addition to using this feature in the predictor, we calculate and test another, more global one: the square root of the squared differences along the sequence, which we will call the ‘normdelta’ score (because it is also the norm of the difference

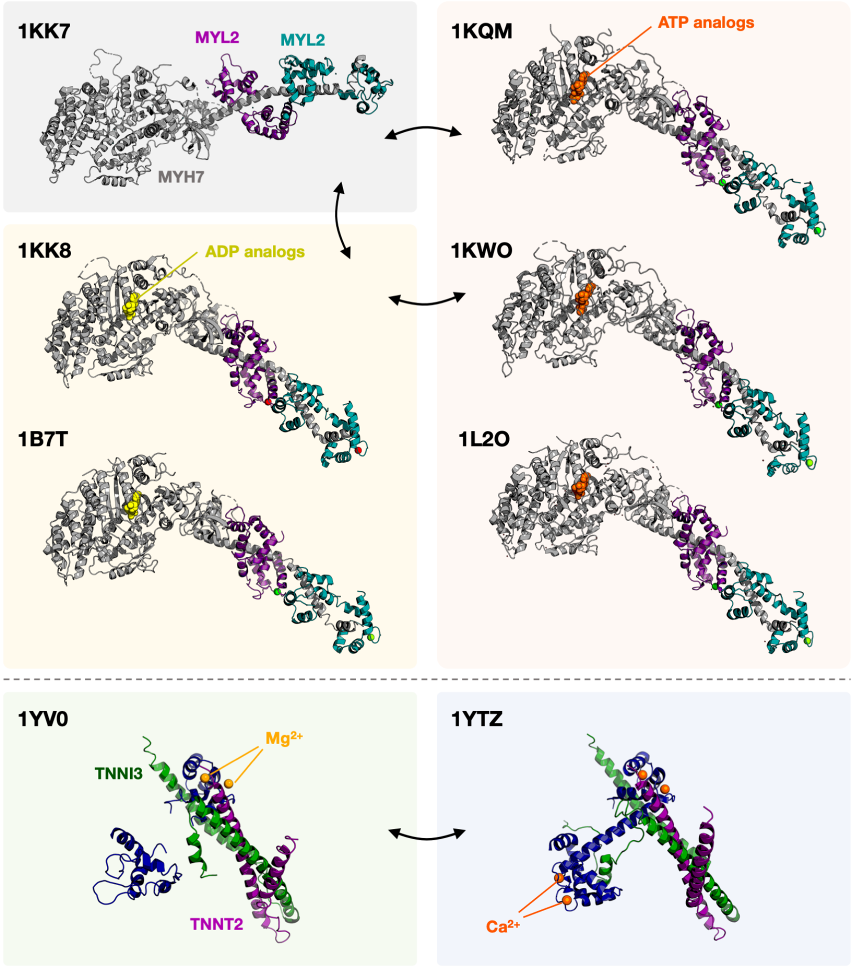


Figure 3.1: **The two sets of PDB folds used for the protein structure comparison score.** The first set includes the scallop myosin heavy and light chains with ADP analogs, ATP analogs, or no nucleotides. The second set includes three troponin chains in the presence and absence of calcium ions.

vector between the COILS scores for the two sequences).

The Protein Structure Comparison Score

The protein structure comparison score (henceforth, the structure score) is a feature that incorporates information about the structural and functional importance of a residue in a protein fold. Jordan et al. try to estimate this importance by aligning the protein folds of the same protein in different functional states and then, for each position in the protein, comparing the distance between the corresponding residues across the two aligned folds with the expected value for that distance. In essence, the structure score represents how much a residue 'moves' after aligning it with the same protein at a different functional state. PolyPhen-HCM assumes that resides that move very little are important for the structural stability of the protein, residues that move a lot are important for protein function, and residues that move some expected distance in the middle are less important. This expected distance is estimated using the crystallographic temperature B factor, also known as the Debye-Waller factor, which is a value proportional to the mean square displacement $\langle u^2 \rangle$ of an atom from its equilibrium position due to disorder inside the crystal lattice:

$$B = 8\pi^2 \left\langle u^2 \right\rangle . \tag{3.2}$$

The observed position of an atom around its equilibrium position is often approximated with a spherical multivariate Gaussian around this equilibrium position and with variance $\boldsymbol{I}_3\sigma^2$, where $\boldsymbol{I}_3$ is the 3×3 identity matrix and $\sigma^2 = \langle u^2 \rangle = B/8\pi^2$.

In PolyPhen-HCM, this method is applied to four of the six genes of interest, namely MYH7, MYL2, TNNT2, and TNNI3. Two sets of structures where used for this task. The first is a set of six structures of a three-chain scallop myosin complex, which includes the myosin heavy chain and two myosin light chains, corresponding to human MYH7, MYL2, and MYL3 respectively. Three of these structures were bound to ATP analogs, two with ADP analogs, and one without any bound nucleotides. The second is a set of two structures of a three chain chicken troponin complex, which includes troponins I, T, and C, corresponding to TNNI3, TNNT2, and TNNC1 in the human heart muscle. One of these structures was bound to calcium ions while the other was not bound to anything (see FIGURE).

Pairwise comparisons were made between folds that represented the same protein in different biologically relevant states (with or without ATP, ADP, or calcium). The structure alignments were made with the software package LovoAlign, and the displacement between the α-carbons of matching residues were measured. Using the variances computed from the B factors and a Student's T-test, a p-value is produced for the observed displacement of each position in the protein fold. To make it such that higher p-values consistently represent a more important residue, values below 0.5 are subtracted from 1.

We built upon several limitations in this statistical analysis. First, while the position of a residue relative to its equilibrium point is sampled from a spherical Gaussian, the distance between said equilibrium point and sample is not Normally distributed; a distance cannot be negative. Therefore, a Student's T-test is inappropriate to calculate p-values. Second, Jordan et al. consider the average of the B factors to calculate the variances. We present a different and more general statistical framework that formalizes the structure score calculations in PolyPhen-HCM. Using the spherical Gaussian approximation of the position of α-carbons in a protein fold, that position can thus be represented as follows:

$$\boldsymbol{X}_i^{(1)} \sim \mathcal{N}\left(\boldsymbol{\mu}_i^{(1)}, \left(\sigma_i^{(1)}\right)^2 \boldsymbol{I}_3\right) \tag{3.3}$$

$$\boldsymbol{X}_i^{(2)} \sim \mathcal{N}\left(\boldsymbol{\mu}_i^{(2)}, \left(\sigma_i^{(2)}\right)^2 \boldsymbol{I}_3\right). \tag{3.4}$$

Where $\boldsymbol{X}_i^{(1)}$ is the position of residue i in the first protein, $\boldsymbol{\mu}_i^{(1)}$ a 3-dimensional vector of means, $\boldsymbol{I}_3$ the 3×3 identity matrix, and $\left(\sigma_i^{(1)}\right)^2$ is the variance (or mean square displacement) of that residue's position in 3 dimensions. The same notation applies to residue i in the second protein. When we align these two proteins, we observe the distance between the two residues, which is the Euclidean norm of the difference vector:

$$\boldsymbol{X}_i^{(1)} - \boldsymbol{X}_i^{(2)} \sim \mathcal{N}\left(\boldsymbol{\mu}_i^{(1)} - \boldsymbol{\mu}_i^{(2)}, \left[\left(\sigma_i^{(1)}\right)^2 + \left(\sigma_i^{(2)}\right)^2\right] \boldsymbol{I}_3\right) \tag{3.5}$$

Assuming that $\boldsymbol{\mu}_i^{(1)} = \boldsymbol{\mu}_i^{(2)} = 0$, it is known that the distribution of the Euclidean norm of a spherical Gaussian is the Maxwell-Boltzmann distribution with the same parameter as the Gaussian above:

$$\left\|\boldsymbol{X}_i^{(1)} - \boldsymbol{X}_i^{(2)}\right\|_2 \sim \mathcal{MB}\left(\left(\sigma_i^{(1)}\right)^2 + \left(\sigma_i^{(2)}\right)^2\right). \tag{3.6}$$

See FIGURE for a simulation of this statistical model, and notice that (i) the Maxwell-Boltzmann distribution has for support only the positive real numbers, and that (ii) it's density is concentrated around 'moderate' values, meaning that relatively small and relatively large displacements are less likely. The latter observation matches the intuition that Jordan et al. rely on when subtracting the p-values from 1 such that less likely displacements (small and large, as opposed to moderate) consistently have larger p-values. We can formalize this using the two tailed hypothesis test in which the null hypothesis stipulates that $\boldsymbol{X}_i^{(1)}$ and $\boldsymbol{X}_i^{(2)}$ are spherical Gaussian-distributed with the same means $\boldsymbol{\mu}_i^{(1)} = \boldsymbol{\mu}_i^{(2)} = 0$. The rejection region for this null hypothesis would be either tail of the Maxwell-Boltzmann distribution. The p-value for each residue i is therefore:

$$p_i = \min\left\{F_{i,\mathcal{MB}}\left(\left\|\boldsymbol{X}_i^{(1)} - \boldsymbol{X}_i^{(2)}\right\|_2\right), 1 - F_{i,\mathcal{MB}}\left(\left\|\boldsymbol{X}_i^{(1)} - \boldsymbol{X}_i^{(2)}\right\|_2\right)\right\}, \tag{3.7}$$

where $F_{i,\mathcal{MB}}(\cdot)$ is the cumulative distribution function of the Maxwell-Boltzmann distribution with the appro-

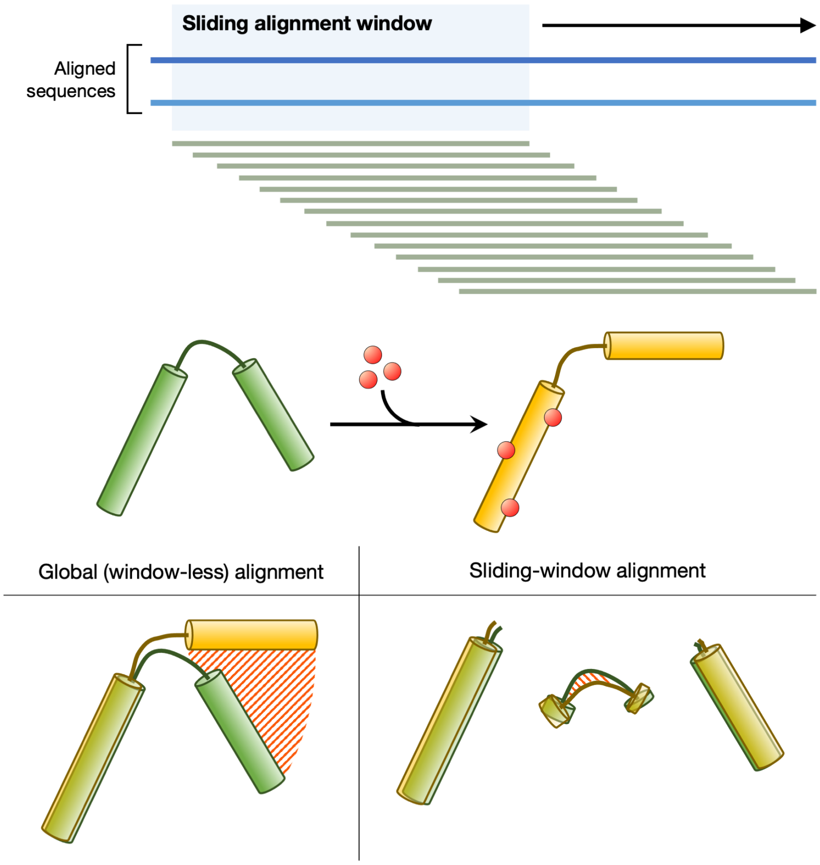


FIGURE 3.2: **Sliding-window alignment.** The rationale behind sliding-window alignment is that we would like to prevent small conformational changes from affecting the global alignment. As seen in the bottom part of the figure, even if the conformational change only affects the loop between the two $alpha$-helices, a global alignment will considerably overestimate the displacements for the smaller α-helix. In essence, a sliding-window alignment is a way to upweight robust local changes over less stable global ones.

priate parameters at residue i.

Because there might be several PDB folds for the same protein in the same biological state, we use Fisher's method to combine the p-values. Knowing that p-values are uniformly distributed, taking their logarithm will yield an exponentially distributed random variable. The sum of many such variables can be represented as a χ^2-distributed random variable. The statistic used in Fisher's method of combining p-values is therefore:

$$-2\sum_{i=1}^{K}\log(p_i) \sim \chi^2_{2K} \tag{3.8}$$

In addition to the statistical model elaborated above, we use a 'sliding-window' alignment to align the protein folds (see Figure **3.2** for a visual explanation). We switched to this alignment scheme because a global alignment can

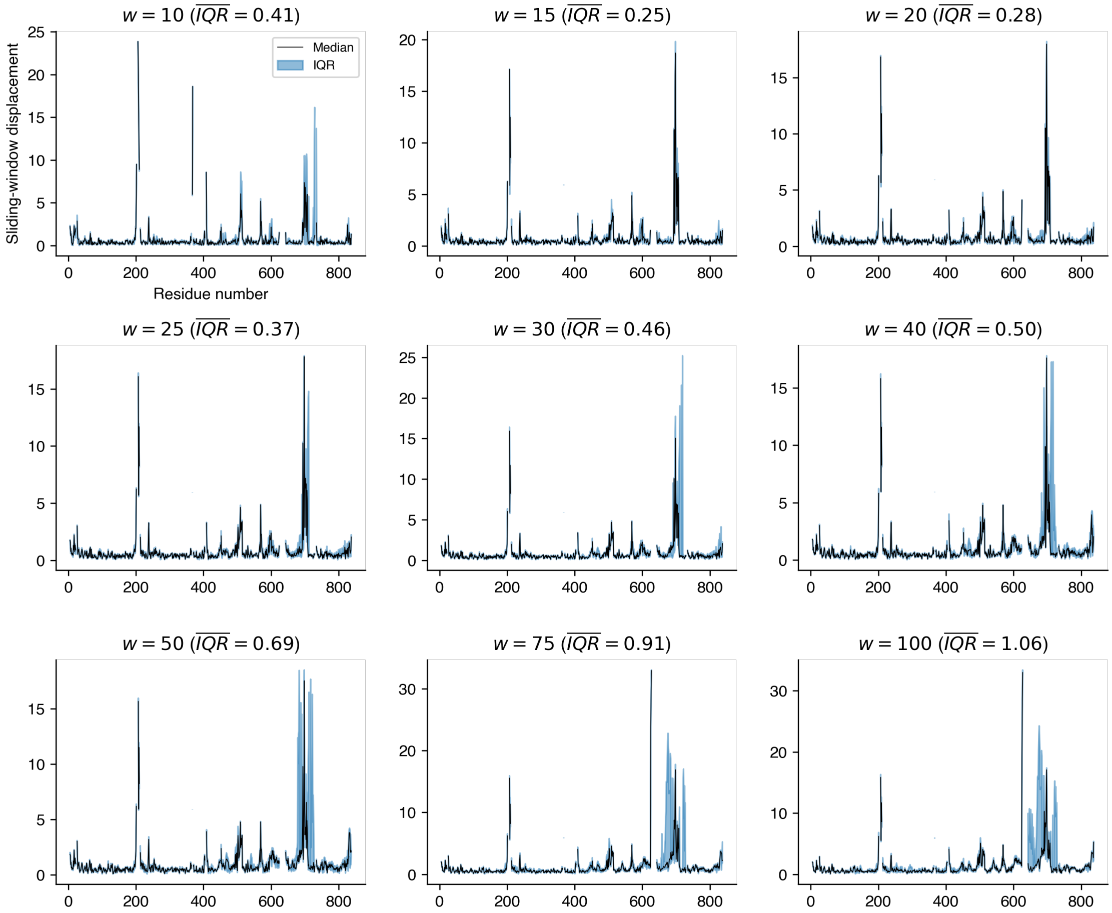


FIGURE 3.3: **Choosing an appropriate sliding-window width.** Sliding-window alignments were performed on the scallop myosin heavy chain (PDB 1kk7) using different window widths, and the median and interquartile range (IQR) were plotted. The window width with the smallest average IQR was chosen, in this case when $w = 15$. Notice how the $\overline{\text{IQR}}$ has a minimum around $w = 15$, as it starts high, then decreases, and increases again. This follow our intuition about the sliding-window alignment; following the Goldilocks principle, a window too narrow will be unstable because of smaller sample sizes, a window too wide will be more susceptible to global changes, a right-sized window is one in which displacement values are most internally consistent.

lead to disastrous misestimations. For instance, the small pivoting of a hinge region linking two protein domains to one another may massively overestimate the relative displacements of residues in the smaller domain (see the lower panel of Figure **3.2** for an example). We noticed this trend when aligning PDB 1KK8 and 1KK7; the displacements of residues in the myosin heavy chain's coiled-coil tail were a lot higher than expected because the global alignment algorithm prioritized the alignment of the much larger globular myosin heads. After the sliding-window alignments are completed, the displacement at a particular residue position is simply the average of the displacements at the same position across the sliding windows that contained it. The width of the sliding window was set to 15 residues because it was empirically observed that it led to the smallest interquartile range of displacements at any position

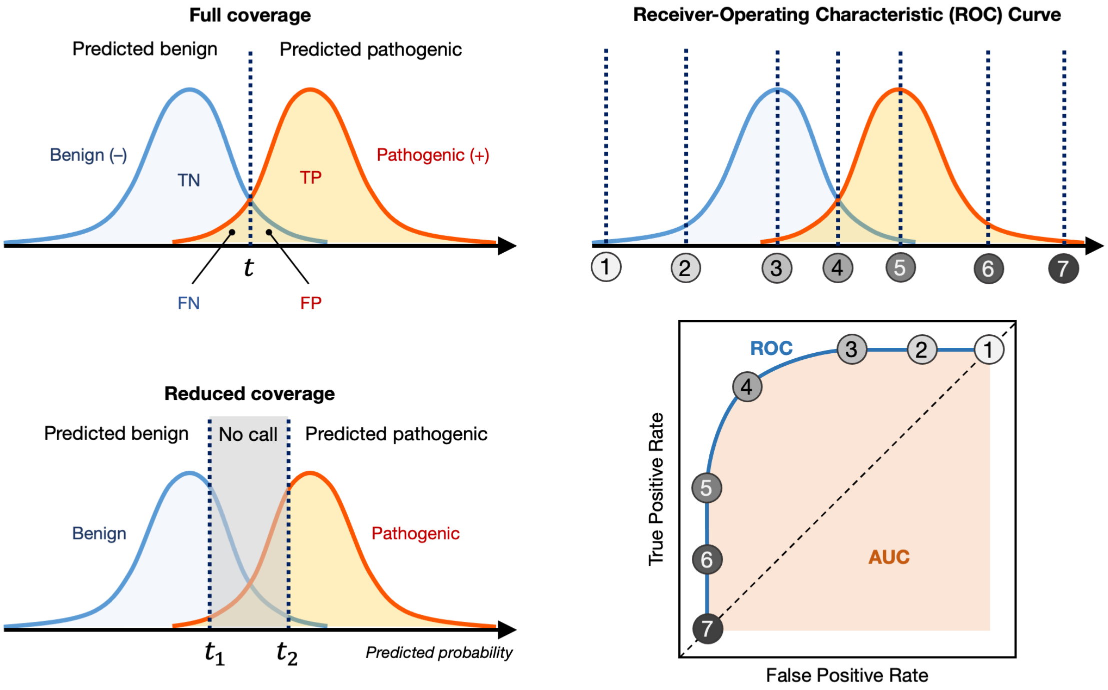


FIGURE 3.4: **Full and partial coverage; ROC curves.** Left: conventionally, a single threshold is used to separate two categories based on their score or prediction probabilities. Alternatively, to increase accuracy an only predict with high confidence, one can introduce a 'no call' region where misclassification happens the most. This would be done at the expense of coverage, as fewer inputs will now be given a prediction. Right: how receiver-operating characteristic (ROC) curves are plotted. ROC curves are incredibly informative performance indicators, as they test model accuracy on all possible thresholds.

across the windows that contain it (Figure 3.3). We confirmed the soundness of this statistical model and alignment approach by verifying that the distribution of all displacements is almost identical to that of a mixture of Maxwell-Boltzmann distributions (data not shown).

Visual inspection of the amino acid residues predicted to be important according to the structure score showed that while many of those residues lied at reasonably important positions (such as the junctions and hinges between adjacent domains of a protein, or the interfaces between secondary structures), many other predictions occurred at superficial and under-constrained loops that can substantially move around in different directions. Interestingly, their large B factors were not large enough to prevent the overestimation of their structural or functional importance (data not shown).

3.1.3 Model Training, Validation, and Feature Ablation

Once the features were ready, I built SVM classifiers using the Python software package `Sklearn`, which implicitly uses LIBSVM, a commonly-used library for support vector machines. To tune the hyperparameters for our model, I used grid search that maximized *balanced* accuracy. Indeed, to account for the relatively small number of benign variants, I weighted benign variants and pathogenic ones such that both classifications has the same sum of weights. For each combination of hyperparameters, 300 shuffle-splits were created, and the combination with the highest average balanced accuracy was chosen. The combinations included RBF and polynomial kernels and their associated parameters, such as the γ factor for RBF kernel and the number of degrees d for the polynomial one. Once the optimal parameters were ascertained, I validated the model by testing it on 1000 different shuffled and stratified test-train splits, which is equivalent to running stratified K-fold cross validation $1000/K$ times and then averaging the results.

To implement the 'no call' classification, I further processed the outputs of the SVM models above. Each model based on a particular train-test split returns not only the prediction for each input, but also an estimated probability that this input's class is the predicted one. The goal was to find two thresholds (like in FIGURE) that maximized accuracy while keeping the coverage above some user-defined minimum coverage. To do so, a brute-force optimization algorithm was used. Briefly, every pair of possible thresholds for the probabilities was tested, and the one that maximized accuracy while keeping a minimum coverage was selected.

To better understand the contribution of each feature, these two steps were repeated for models that included all or all-but-one of those features (yielding 5 combinations in total). And for different feature combinations and minimum coverages, the confusion matrix, the receiver-operating characteristic (ROC) curves, and the area under the ROC curve (AUC) were computed using different `Sklearn` submodules. Confidence intervals for sample averages and standard deviations were determined using the Bootstrap method.

Because the COILS and protein structure comparison scores are only available for a subset of the genes of interest, there is a lot of missing data in in those two features. To address this issue, I use `Sklearn`'s multivariate feature imputation, which is an method that models missing values as functions of other features and uses the fitted regression models for iterative imputation. Using such a complex imputer imposes many assumptions about the structure of the data and the relationships among its features, but we thought it might be more informative than replacing a missing value with the mean value of a column. In addition to fancy imputation, I added a column

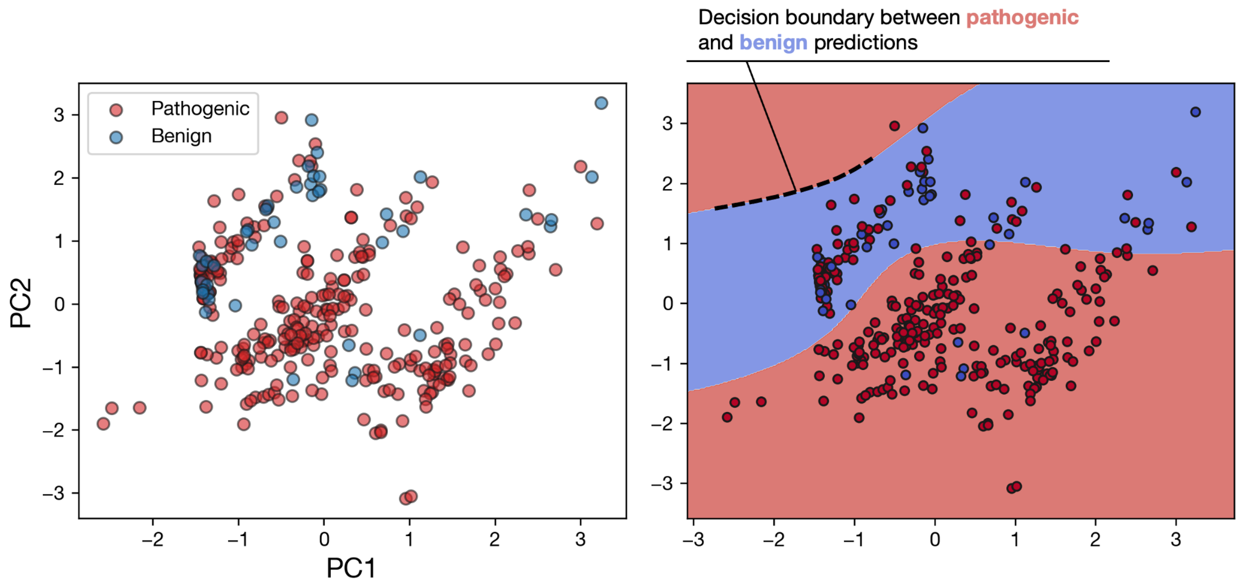


FIGURE 3.5: **Visualizing the data and the classifier in two dimensions.** Using principal component analysis (PCA), one can reduce the dimensions of the 4-dimensional data into 2 dimensions for exploratory purposes. The plot on the right is made by creating a 2-dimensional grid and evaluating the classifier at every point on that grid. The decision boundary (in 4 dimensions) between pathogenic and benign variants is what the learning algorithm will estimate.

of indicator variables for each feature where there might be missing data (e.g. COILS and structure comparison). This indicator variable for an input would be 1 if the corresponding feature is missing, and 0 otherwise.

3.2 Results

3.2.1 An SVM classifier for HCM variant classification

Using conservation, evolutionary, and structural information, I redesigned an algorithm for the classification of HCM-specific missense variants occurring in either one of 8 proteins central to the the function of the cardiac sarcomere. PolyPhen-2, which is in itself an estimate of deleteriousness, combines more structural and conservation metrics in its regression. The COILS and structure comparison scores provide protein-specific information about the coiled-coil tendency and functional importance of a residue position. The MrBayes-computed substitution rate is another covariate that contributes more phylogenetic and sequence conservation information. Starting from a set of annotated variants, the 4 features are computed using the ENSEMBL Variant Effect Predictor (VEP), the NCOILS, LovoALign, MrBayes software packages, and the statistical methods and sequence alignments that use the outputs of those programs for feature engineering.

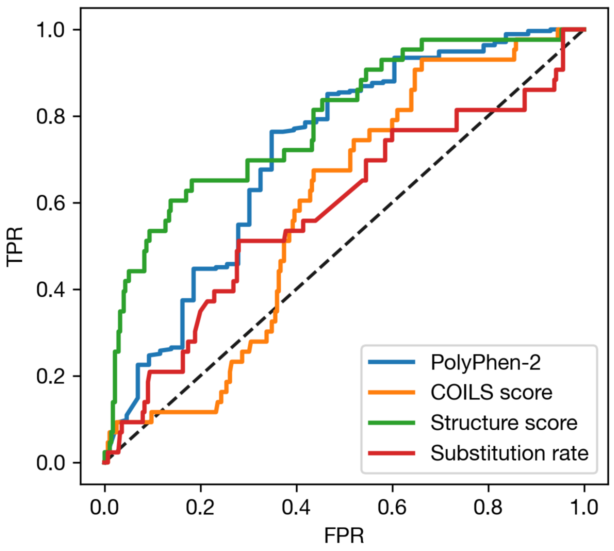


FIGURE 3.6: **Diagnosing trends with a ROC per feature.** Plotting the ROC curves of each feature independently can provide useful intuition on which feature(s) contribute the most to the accuracy of a classifier. Here, the structure score seems to have the largest AUC, which explains the drop in accuracy of the ΔStructure score model in Figure **3.7**. The COILS score has the lowest AUC, which explains the increase in accuracy of the ΔCOILS score in the same figure.

Based on these features and known variant labels, an RBF kernel SVM model is trained to learn a decision boundary separating the variants to be predicted as pathogenic from the ones to be predicted as benign. To illustrate this, I used principal component analysis (PCA) to project the 4-dimensional data onto a 2-dimensional hyperplane, and trained an RBF kernel SVM model with similar parameters on this 2-dimensional data. Figure **3.5** displays the results of this modeling. On the rightmost plot, any variant that falls into the blue region will be predicted as benign, and any variant that falls into the red region will be predicted as pathogenic. Because our classifier using a (non-linear) RBF kernel, the boundary between the two regions is not a line; its complexity depends on the hyperparameters of the kernel. At this stage, one can already notice that the benign and pathogenic variants (blue and red on the left graph, respectively) are not easily separable, as they can cluster together. Such poor separability might have resulted from inherent imitations in the features or from other parts of the analytic workflow.

3.2.2 Model accuracy increases with decreasing coverage

Table **3.1** and Table **3.2** show the average accuracy rates for each experiment, as well as the standard deviation of accuracy scores to allow the comparison of the consistencies across different models and settings. First, I performed a feature ablation experiment where in the SVM is trained on either all or all-but-one features (5 combinations in

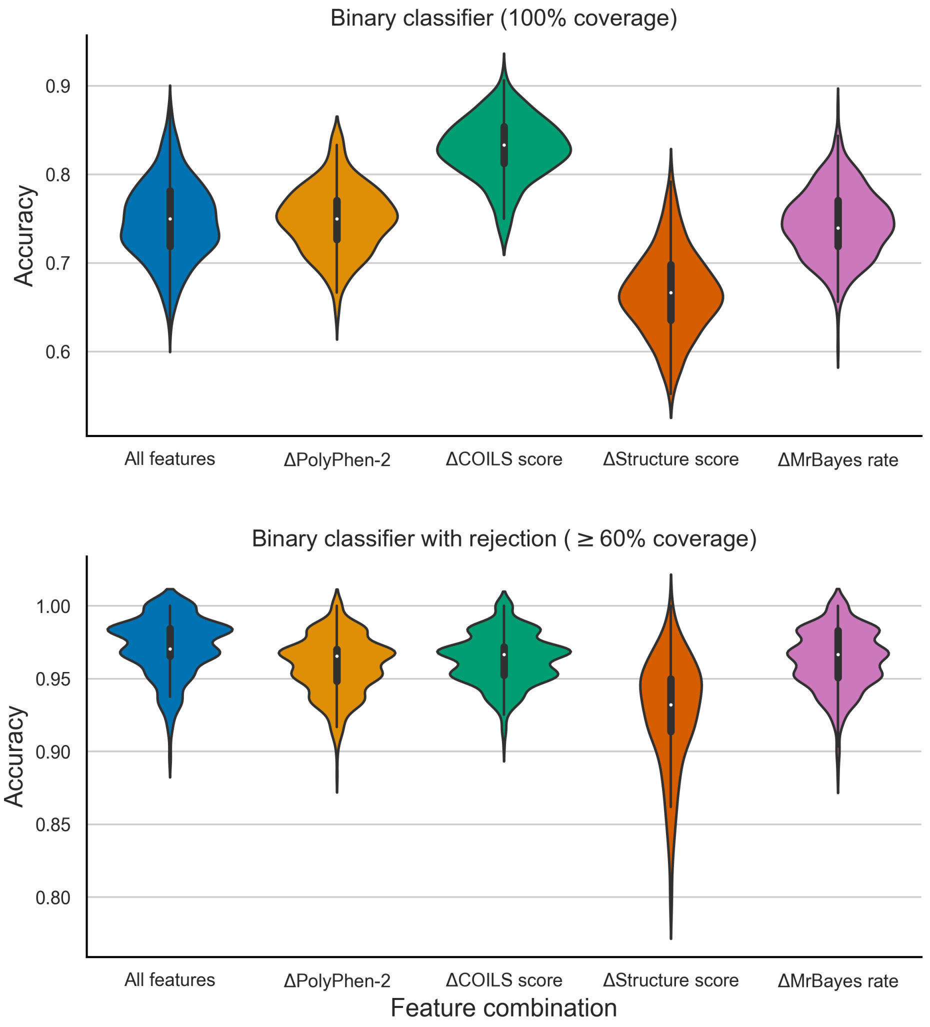


Figure 3.7: **Model accuracy, feature ablation, and coverage.** For each feature combination and at two coverage levels, 500 train-test splits of the data were produced, and an RBF kernel SVM was trained and tested on each one of those splits. The violins are simply vertical density estimators of the distribution of the scores of each model on each train-test split. I chose to measure performance using stratified shuffle splits because they are more general than stratified K-fold cross-validation, since the latter depends on the order of the data. The accuracy trends in feature ablation can be explained in Figure **3.6**. As expected, reduced coverage increases accuracy.

total, see the topmost violin plots in Figure **3.7**). The performance here matches that of Jordan et al. very closely; with 100% coverage, the average accuracy is around 75% for all features, and and it remains very close that in the ΔPolyPhen-2 and Δsubstitution rate feature combinations. However, in a manner also consistent with the original paper, omitting the COILS score seems to increase accuracy by a little bit and omitting the structure score seems to make it worse. It is not understood exactly why this happens, and the fact that all the missing values occur in either of those features adds another layer of complexity, as about 40% of the variants have no COILS or structure score on average.

One possible way to diagnose this phenomenon is to look at the receiver-operating characteristic (ROC) curves for each feature and see how it would perform as a predictor variable on its own. ROC curves, and the area under them, are a useful way to quantify the degree of separability between pathogenic and benign inputs. Looking at Figure **3.6**, it is clear that the structure score is by far the feature that best separates pathogenic from benign variants, while the COILS score is the worst. While this diagnostic only tells us about each feature independently of one another (as opposed to all the features in 4-dimensional space), it provides us with intuition about which is going to be most or least useful to start with in the classification task.

Moreover, I processed each SVM classifier from the feature ablation experiment such that it can reject a variant if it falls within an accuracy-maximizing rejection region that predicts for at least 60% of the inputs. Again, the results almost perfectly match those in the original paper: all classifiers perform considerable better at 60% coverage compared to full (100%) coverage. This time, ablating the COILS score feature does not lead to an increase in accuracy, but ablating the structure score does lead to a slightly but significantly worse and less consistent average accuracy (see bottom-most violin plots in Figure **3.7**). This result also matches that in Jordan et al. (2011).

The reason accuracy increases as the minimum coverage drops is because the rejection region tends to contain ranges where the probability distributions for each category overlap, and that is where most of the misclassification happens. Figure **3.8** neatly shows this trend: at first, a steady increase, until the limit is reached, with most scores falling in the 90-100% range when minimum coverage is at 70%. The bottom histogram plot of the same figure shows why we say 'minimum' coverage: the model might perform better on a train-test split than on another, and since the rejection region can be smaller with better splits, some splits might necessitate smaller rejection regions than other ones. However, this only happens with a minority of splits, as demonstrated in that same figure.

3.2.3 The side effect of decreasing coverage

Decreasing coverage and only making high-confidence predictions seems to be a compelling trade-off in variant pathogenicity prediction. However, there is more to it than meets the eye. Enlarging the rejection region does not only trivially lead to a lower coverage, it also leads to a much larger variance in ROC-AUC performance — especially in an unbalanced problem like this one. ROC curves and their AUCs are important indicators of performance, for they are insensitive to class balance. Accuracy, on the other hand, is highly sensitive to it, as it is based on a single threshold, whereas ROC curves evaluate a model over all possible thresholds and is therefore a very informative measure in all balancing settings.

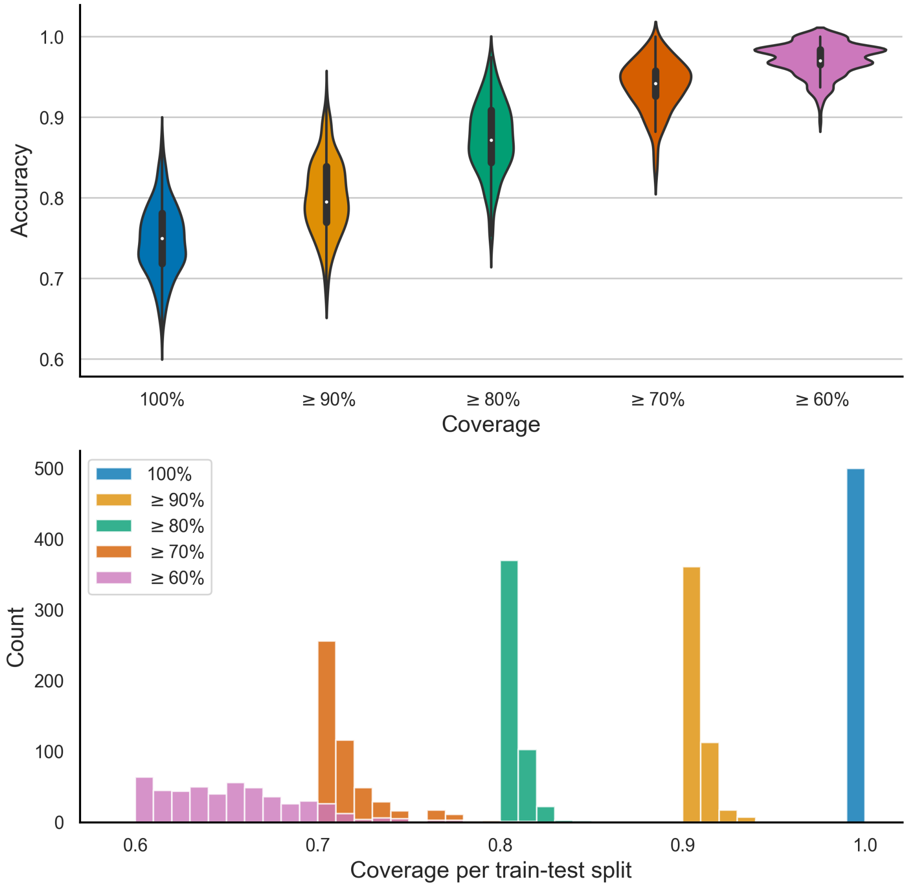


Figure 3.8: **Decreasing coverage increases accuracy.** Top: as in Figure 3.7, the violin plots are density estimators of distribution of scores over 500 train-test splits of the data. The smaller the minimum coverage, the greater the accuracy. This is because the 'no call' region gets wider and excludes more and more potential misclassifications. Bottom: a minimum coverage is only a lower bound on the allowed coverage, it does not mean that all train-test splits will have exactly this minimum coverage. This histogram shows the distribution of coverages among the splits at different coverage minima. The colors between the top and bottom plots are related by that minimum coverage.

Figure 3.9 shows how ROC performance changes as we lessen the minimum coverage. The top-most level confusion matrices shows that while pathogenic recall (e.g. the proportion of pathogenic variants that are predicted to be pathogenic) increases, benign recall actually shrinks. And while variance in pathogenic recall dwindles, variance in benign recall grows quickly. The middle and bottom-most levels show the same trend affecting the ROC-AUC statistics: the lower the minimum coverage is, the higher the variance of the ROC and its corresponding AUC becomes.

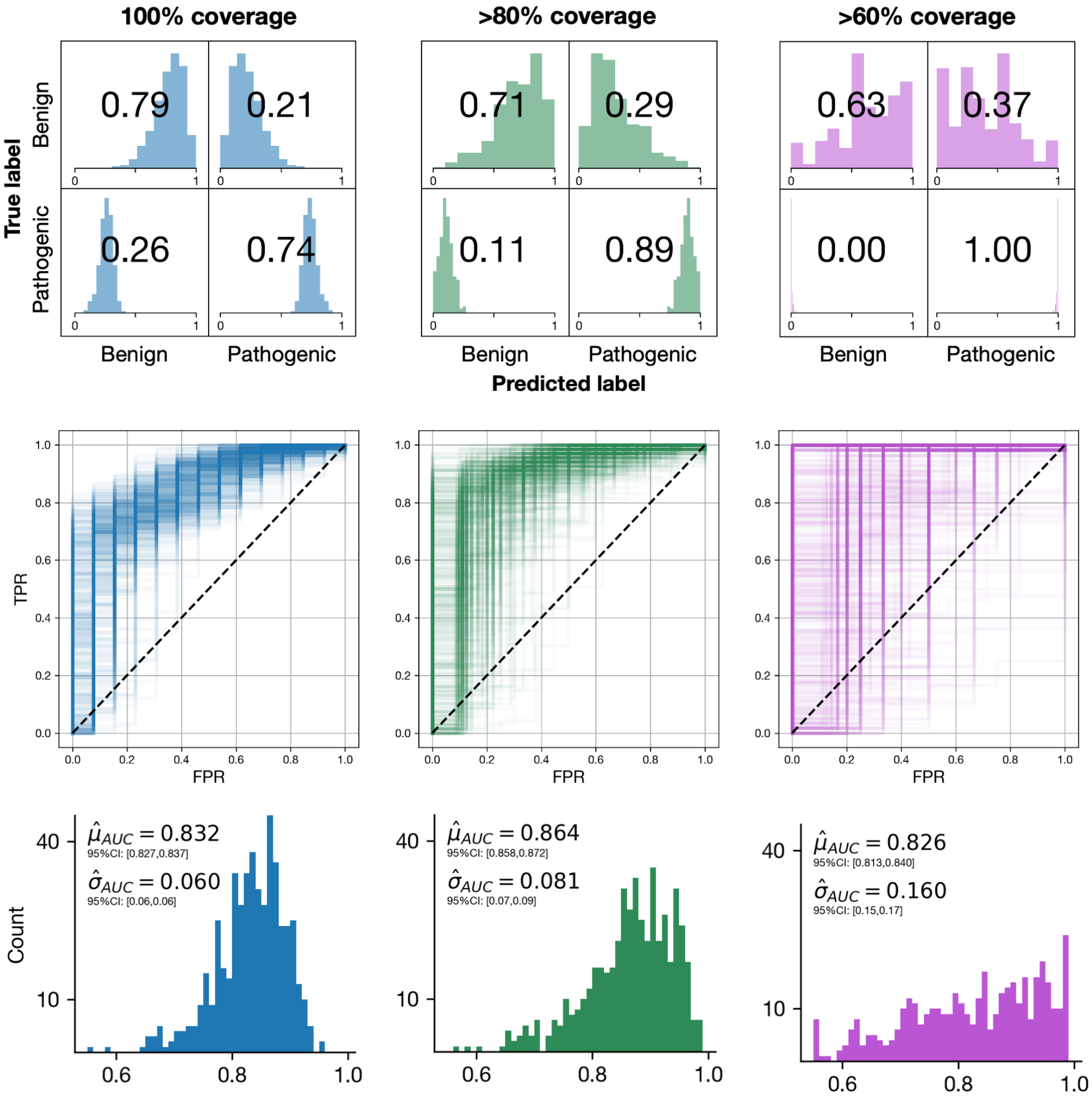


FIGURE 3.9: **The side effects of decreasing coverage.** For 100%, 80%, and 60% minimum coverage, three performance indicators are shown in the corresponding column. Top: to evaluate performance more holistically, the distribution of row-normalized (the rows sum up to 1) confusion matrices was plotted as a histogram within each corner of the confusion matrix. Notice that while pathogenic recall increases as coverage is decreased, benign recall decreases in average and increases in variance. Middle: the ROC curves of all 500 train-test splits, superimposed. As coverage increases, ROC curves become more variant. 80% minimum coverage seems to be close to an optimal coverage where accuracy is higher and variance is acceptable. Bottom: to show the variance of the ROC curves in the middle section, the distribution of their AUC values was estimated with histograms. These confirm our observation that 60% coverage lead to high variation in performance level, and that 80% coverage is close to an optimal coverage that optimizes accuracy and AUC variance together.

However, this increase in variance and imbalance in recall is much more significant when the minimum coverage is at 60% than when it is at 80%, indicating that there is an optimal middle point at which this 'coverage-variance' trade-off is acceptable. The mean AUC actually increases with accuracy at 80% minimum coverage (with a modest

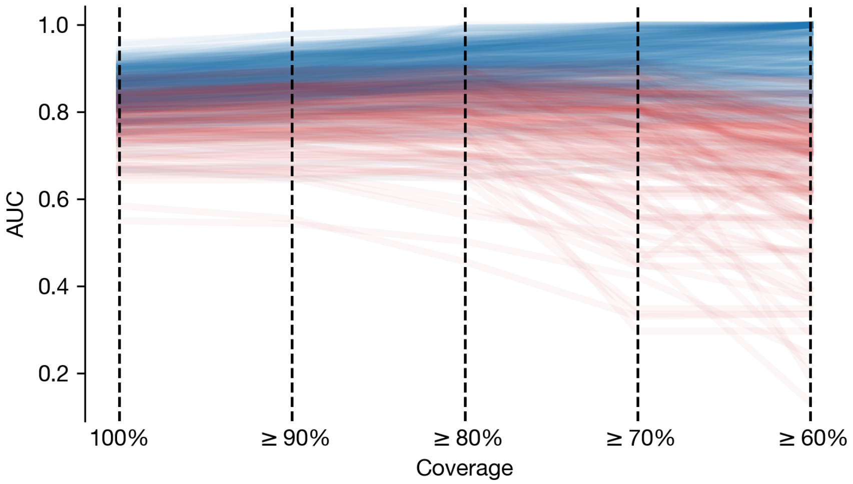


FIGURE 3.10: **Reducing coverage makes good AUCs better and bad AUCs worse.** Each line represents the AUC of each one of the 500 SVM models trained and tested on one of the 500 train-test splits (all 4 features included). Red lines see their AUC decrease from 100% to 60% minimum coverage and blue ones see their AUC increase. This confirms our intuition that reducing coverage makes high-AUC models better and low-AUC models worse.

increase in variance). And while many of the classifiers at 60% minimum coverage have an accuracy of 100%, almost half of the remaining ones have an accuracy lower than 80%, compared to a much lower proportion at higher minimum coverages. This result also suggests that there is a critical point between 100% and 60% coverage (perhaps somewhere around 80%) at which this substantial surge in variance happens and the AUC starts to decrease. This result is not surprising; with a data set size on the order of 300 variants and a notable imbalance between the two classes, there must be a rejection region beyond which very few benign variants are actually classified, which could easily escalate the variance of the AUC (data not shown). By a similar and simple rationale, I hypothesize that adding a 'no call' category makes good classifiers better and bad ones even worse. Figure **3.10** confirms this hypothesis.

Throughout the analysis above, we realize this additional trade-off between coverage and ROC-AUC, the latter of which is a much more instructive and explanatory measure than accuracy alone. This effect was not discussed in Jordan et al. (2011), though it is of prime importance when it comes to clinical significance. Without meticulous evaluation, imbalanced data — which is not unusual in clinical genomics — could lead to compromised classifiers with similarly imbalanced performance and a potentially harmful clinical impact. In this case, the imbalance is in favor of pathogenicity: the classifiers we learned predict pathogenic variants better than benign ones, and it is up to us to decide whether overestimating the pathogenicity of benign variants is something that we can tolerate in the

	Accuracy			
	100% coverage		≥60% coverage	
Feature combination	Mean	SD	Mean	SD
All features	0.748 [0.744 - 0.752]	0.044 [0.041 - 0.046]	**0.972** [0.97 - 0.974]	0.02 [0.019 - 0.022]
ΔPolyPhen-2	0.75 [0.746 - 0.753]	0.038 [0.035 - 0.04]	0.959 [0.957- 0.961]	0.02 [0.018 - 0.021]
ΔCOILS score	**0.828** [0.825 - 0.831]	0.034 [0.032 - 0.036]	0.963 [0.962 - 0.965]	0.017 [0.016 - 0.018]
ΔStructure score	0.666 [0.661 - 0.67]	0.046 [0.043 - 0.049]	0.927 [0.924 - 0.93]	0.037 [0.035 - 0.04]
ΔSubstitution rate	0.745 [0.741 - 0.748]	0.038 [0.036 - 0.041]	0.963 [0.961 - 0.965]	0.02 [0.019 - 0.022]

Table 3.1: **Sample averages and standard deviations of the accuracy of each model in the feature ablation experiment.** Bracketed values are the lower and upper bound of a 95% confidence interval computed using the Bootstrap method.

clinic.

3.3 Discussion

In this chapter, I presented the work I did to reproduce, formalize and make available PolyPhen-HCM's underlying algorithms and statistical methods to the public. This included creating a code base to compute the features and learn the parameters of a support vector machine from them, developing an empirically supported and first-principled statistical model to identify structurally or functionally important residues in a protein based on its folds at different biochemical states, and devising and using a consistent sliding window strategy for more robust fold alignments. I then applied this framework to the most recent variant data available, showing that its performance hasn't changed and that there is much more to coverage reduction than an almost-free increase in accuracy. At the same time, I engaged different questions and paradigms in pathogenicity prediction, such as bias, data balance (or the lack thereof), scarce data, variant labeling, measures of accuracy, among other things.

While I was able to replicate PolyPhen-HCM, improving it significantly using the same features proved to be a challenging task. The observation that new data and redesigned methods do not improve classification significantly is an important one that further speaks to the challenges that PolyPhen-HCM is trying to address. This can be for many reasons, including the shortcomings that I mentioned at the beginning of this chapter and later on as

	All features			
	Accuracy		AUC	
Coverage	Mean	SD	Mean	SD
100%	0.748 [0.744 - 0.752]	0.044 [0.041 - 0.046]	0.832 [0.826 - 0.837]	0.06 [0.055 - 0.065]
90%	0.801 [0.797 - 0.805]	0.047 [0.044 - 0.049]		
80%	0.874 [0.87 - 0.878]	0.045 [0.043 - 0.048]	**0.864** [0.857 - 0.872]	0.081 [0.073 - 0.088]
70%	0.939 [0.936 - 0.942]	0.033 [0.03 - 0.035]		
60%	**0.972** [0.97 - 0.974]	0.02 [0.019 - 0.022]	0.826 [0.812 - 0.84]	0.16 [0.146 - 0.173]

Table 3.2: **Sample averages and standard deviations of the accuracy of each model in the coverage reduction experiment.** Bracketed values are the lower and upper bound of a 95% confidence interval computed using the Bootstrap method.

we discussed the data such as persistent data imbalance due to ascertainment bias against benign variants, poor or incorrect clinical labeling, the inadequacy of the hypothesis class implied by the RBF kernel SVM, or a combination of all these potential issues.

Notwithstanding these issues, PolyPhen-HCM is still a powerful proof-of-concept, for it integrates different kinds and sources of information such as evolutionary conservation and domain knowledge about protein structure in an attempt to understand the relationships and correlations among these features and learn a way to predict pathogenicity from them. Within a few years of the development of this method, CADD, also an SVM that combines even more features, was published. Of course, CADD's approach is drastically different from that of PolyPhen-HCM, as it circumnavigates the issues of ascertainment bias and inaccurate labeling by simulating variants and comparing them with observed ones. However, its learning approach is still quite similar to PolyPhen-HCM: both use an SVM and a comprehensive assortment of carefully engineered features for prediction. Another unique aspect of PolyPhen-HCM is that is is specifically tailored for variants that occur in one of a few disease-relevant genes. As far as we know, no other published method uses a protein structure comparison score for the task of classifying variants, and as it turns out, this score is the most important predictive feature in the model.

As we saw in Chapter 2, the methods that followed steadily increased in the complexity of their models, slowly including more and more hidden variables to improve prediction. With that, they develop frameworks that neither

PolyPhen-HCM, CADD, nor any similar method possesses: a generative model. Indeed, SVMs are only discriminative; they are supervised in nature and are used for classification or regression purposes. Contrastingly, the more complex Potts and variational autoencoder models optimize the parameters of the evolutionary generative distribution that produces the sequences observed in the population. Once this distribution is approximated, and with the assumption that is is correlated with fitness, it is used to infer the probabilities of those sequences and their relative fitness and pathogenicity. Generative models are extremely appealing, not only because of their capabilities once they are fine tuned, but because they can be grounded in intuitive principles and representations of the world.

In the upcoming chapter, I propose a new, but related, generative model along with a series of other contributions to improve its architecture, training, coverage, and generalizability. Just like the evolutionary couplings and VAE frameworks, it incorporates in its optimization myriads of hidden variables (which can be thought of as variables in the causal network producing the data). However, it also proposes novel solutions to a few long-standing issues.

Essentially, all models are wrong, but some are useful.

George E. P. Box

4

An Interactome-Based Deep Generative Model for Pathogenicity Prediction

Building on our review of foundational methods in Chapter 2 and our detailed treatment of a main baseline method for pathogenicity prediction in Chapter 3, we now introduce new approaches to improve pathogenicity prediction by learning richer representations of sequence data and interactions with deep learning approaches. Throughout this chapter, I will recapitulate some limitations of extant models and propose ways of addressing these weaknesses. I will introduce different state-of-the-art deep learning architectures and learning paradigms in order to not only improve performance to be increasingly safe and acceptable in the clinic but also open up avenues for methods that might enable a better understanding of biology and the complex interactions that govern it. Some of these architectures will be based on a generative model to increase interpretability, while others will be based on a purely discriminative one to maximize the accuracy of the task for clinical deployment.

4.1 Motivations

So far, we have pointed out a plethora of issues, assumptions, and limitations that ought to be confronted when thinking about methods for variant pathogenicity prediction. For instance, we have discussed how the complexity of a model affects accuracy and how the representativeness of sequence data affects learning. Here, we focus on three major and recurrent issues that the models hitherto explored have yet to address more coherently: insertions-deletions that change the length of a sequence, integrating interactome information, and working with data that satisfies the i.i.d assumption.

4.1.1 Accounting for variable-length sequences

Dealing with variable-length inputs is a challenging task because it makes the sequence space infinitely large; there are 20^L possible protein sequences of constant length L, but there is an infinite number of possible protein sequences when L is allowed to vary. And indeed, evolution does not only operate through single and multiple nucleotide polymorphisms; it may add or remove residues from a protein to optimize its contribution to organism fitness. It is easy to see where the challenge lies: after having seen sequences of only one length, how does one predict anything about a sequence that is slightly shorter or slightly longer?

Naturally, there are many heuristics that can be applied. SIFT Indel uses covariates such as the length of the insertion-deletion ('indel'), its position relative to the start of a gene, and other features pertaining to the gene and the substituted amino acids to derive classifiers of moderate accuracy (Hu and Ng, 2012). Protein Variant Effect Analyzer (PROVEAN) uses the BLOSUM62 alignment score between a reference sequence and a mutant one to predict the effects of any type of substitution, including indels, since any two sequences can be aligned and scored (Choi et al., 2012), with accuracies ranging from high seventies to middle eighties depending on the type of mutation. However, these methods are limited by scarce data and moderate performance, and the heuristics are lacking in biological context.

The more complex generative models we have discussed in Chapter 2 are also vulnerable to this variable-length issue. In the case of the Potts model and the EC framework, a matrix of interaction energies $\boldsymbol{J}_{ij}$ is learned for each pair of positions (i, j). Therefore, any variation in sequence length or any small shift in residue positions will confuse the energy-based model, assuming that the input sequence is truncated to the correct length. In the case of DeepSequence, each amino acid is one-hot encoded; that is, it is turned into a vector of length 20 where all elements are 0 except the element corresponding to this amino acid, which is 1. (One-hot encoding is the most common and reasonable way categorical variables are encoded in machine learning, see Figure **4.2** and Figure **4.4**.) The one-hot vectors are then concatenated to form a long column vector of length $20 \cdot L$, where L is the length of the protein sequence. These long vectors are the inputs to the DeepSequence VAE, whose architecture — as with any neural network — is only compatible with fixed-length inputs. In a neural network, nodes 'learn' to receive the inputs they receive. Accordingly, any shift or change in sequence length will throw off the inference.

There are sensible workarounds to deal with variable-length inputs as long as a program can deal with gaps. For example, DeepSequence and EVcouplings first create a multiple sequence alignment (MSA) with the sequences in their data, which returns a block of *gapped* fixed-length sequences that will be used for training and optimization. The block sequences will be at least as long as the reference sequence, and they will usually be longer; that length is determined by the longest homologous sequence in the set (say L'). With this in mind, the Potts model and VAE network can be designed to take as input gapped sequences of identical length L'. Still, there are a couple catches: (i) these models risk to overfit the gapped regions and not generalize well for all possible gap positions and gap lengths, and (ii) they will still not account for sequences that are even longer than the longest sequence in the training set. In addition, the choice of alignment algorithm and scoring matrix will likely have a strong impact on the training MSA and the resulting classifier.

4.1.2 Integrating the interactome

The interactome is the map of all interactions between protein, DNA, metabolites, small molecules, inorganic molecules, and physical factors that underlie all biological processes. Signaling pathways, transcription cascades, protein complexes, and RNA-mediated regulation are all included in the interactome. Understanding such an enormous map would facilitate the elucidation of genetic mechanisms behind inherited diseases, the discovery of drug targets and treatment strategies, and our apprehension of epistasis. Its importance has been highlighted with the emergence of precision medicine (Maron et al., 2021) and network-based medicine (Barabási et al., 2011). It is also possible to better understand the relationships among different diseases by analyzing the overlap of their sub-networks, or modules, in the interactome (Menche et al., 2015).

None of the methods considered as yet incorporate interactome information. Although some work has been done to extend the EC framework from intra-protein residue contacts to inter-protein complex residue contacts (Hopf et al., 2014). In this study, the sequences of two interacting proteins were concatenated with each other and treated as a single sequence for direct coupling analysis (e.g. using the Potts model described in Chapter 2). However, the disease networks we are interested in can be much larger than just two proteins, and the variant sequences have to be from the same individual to guarantee that they interact with each other and that they coevolved together. Otherwise, epistatic signals would become much harder to detect.

4.1.3 Issues with Data

An implicit assumption paramount to all methods that use sequence conservation and covariation to predict the effects of variants is that the sequences are independent and identically distributed (the i.i.d. assumption for Section 2.1). That is, the sequences are independently sampled come from a common evolutionary process. This entails that they all be orthologous to each other (as opposed to paralogous) and have identical function. Orthologous genes are related by vertical descent in a common organism and have the same function in different species. Oppositely, paralogous genes are related horizontally and have evolved via gene duplication; they encode proteins with similar, but not identical function. Paralogous genes may therefore not be generated from the same evolutionary processes, and their sequence distribution might be very different from that of orthologous genes with identical function. This assumption is crucial for generalizability, but it is also very hard to satisfy.

Most of the methods mentioned in this thesis use homologous sequences from diverse sets of taxa even though there is no guarantee that these sequences arise from the same generative distribution or evolutionary process. Even if the sequences themselves are orthologous and may seem to have the same function, the same might not be true of the networks of interactions that contain them. Once again, proteins do not evolve independently, they co-evolve with other proteins and nucleotide sequences – at least the ones with which they physically interact. (Somehow, this data problem has become tied to our knowledge of the interactome just as much as the evolution of a sequence is tied to the evolution of the interactome in which it acts.)

That said, because our knowledge about the human interactome is incomplete (let alone that of other species), discovering and learning precise epistatic mechanisms might require more stringent assumptions about sequence data than orthology.

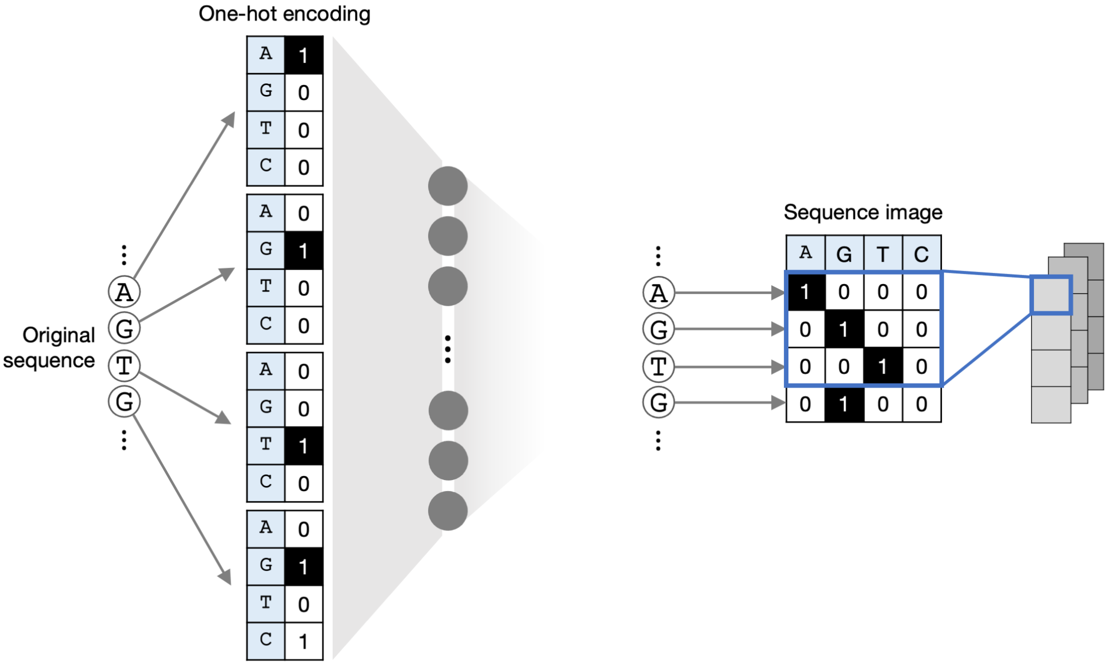


Figure 4.1: **Encoding a sequence into a one-hot 'image'.** Left: how DeepSequence and PEVAE (Section **2.6.3** and Section **2.6.4** respectively) transform sequences into long concatenated one-hot vectors. These vectors are then used as inputs to the VAE encoder network. (The grey trapezoids and the dark grey nodes depict the connections and neurons of the neural network, respectively.) Right: my approach of stacking the one-hot vectors into images that can be used as inputs for convolutional networks with biologically-inspired kernel dimensions. The advantage conferred here is a translational invariance with respect to the shifted domains.

4.2 Proposed Solutions

Motivated by the above, I propose a solution for each problem, and combine these solutions into deep learning models for variant pathogenicity prediction.

4.2.1 Convolutional networks for sequence 'images'

The first problem I deal with is that of variant-length sequences and insertion-deletions. Say an amino acid was inserted somewhere in a protein, causing a right-ward shift for all the ensuing amino acids. We would like our method to be able to still accept this elongated input and not be 'fazed' by the sudden shift in positions. That is, we want our method to still 'read' the shifted input as if the insertion was not there, while nonetheless considering the local change that the insertion causes. There, our method should base its decision on whether such a pattern change was observed in the training data or whether the insertion disrupts another conserved pattern, as in the pathogenic case.

This problem can be formulated as *translational variance*, or contrastingly as the lack of *translational invariance*. A function is translationally invariant if its output does not change when the input is translated (i.e. shifted right, left, up, or down). For instance, a picture of a cat remains a picture of a cat even if the cat was moved from one side of the frame to another. Here, the input is an image and the function maps this image to a category (e.g. 'cat' versus 'not cat'). In the same way, a protein domain is still the same protein domain even when it is shifted by a few residues upstream of it, like in the case of an elongated linker between two domains.

The way this problem is solved with pictures of cats is by using convolutional neural networks (CNNs) that learn the small- and large-scale filters necessary to identify a cat image from something else. CNN filters are also known as kernels. (See Figure **4.2** for a brief visual explanation of convolutions.) I propose we do the same for sequences. Once each nucleotide or amino acid in a sequence is one-hot encoded, the one-hot vectors are stacked to form an $L \times K$ binary image, where L is the length of the sequence and K the number of columns in the sequence image, with $K = 4$ in the case of DNA and $K = 20$ in the case of protein sequences. The image is binary because it consists of only 1's and 0's, and each row contains exactly one 1 (the 1 corresponding to the amino acid or nucleotide at that position). Instead of the classic square kernel dimensions generally used in image classification, I propose we use kernels of dimension $M \times K$, where M is an arbitrarily chosen height, and K is again the number of columns in the sequence image. This way, a convolution will simply be a single downward pass, as if the kernel was scanning the sequence to find some pattern.

This kernel design prevents the CNN from behaving like it would with images of cats. Fancier and more complex CNN architectures (along with data transforms and data augmentation) can recognize cats even when they are flipped or rotated. We do not want rotational invariance in our classifier, and as long as the training sequence data is neither flipped nor reversed (or transformed in any biologically unrealistic manner), there is no chance this will happen with such a kernel and simple CNN design.

That a CNN can still recognize a shifted protein domain does not mean that it will ignore what caused the shift. Its final output will still depend on the added or deleted nucleotides and whether the new pattern is found in the data. The cat image analogy should not serve as a source of intuition here because cat image classifiers are trained on much more abundant and diverse data, whereas variant classifiers would be trained on relatively untouched, highly similar, and well-conserved sequences.

A major advantage of this strategy is that it does not require any multiple sequence alignment of the training sequences to be made. Sequences shorter than the longest sequence can simply be padded with 0-only rows at

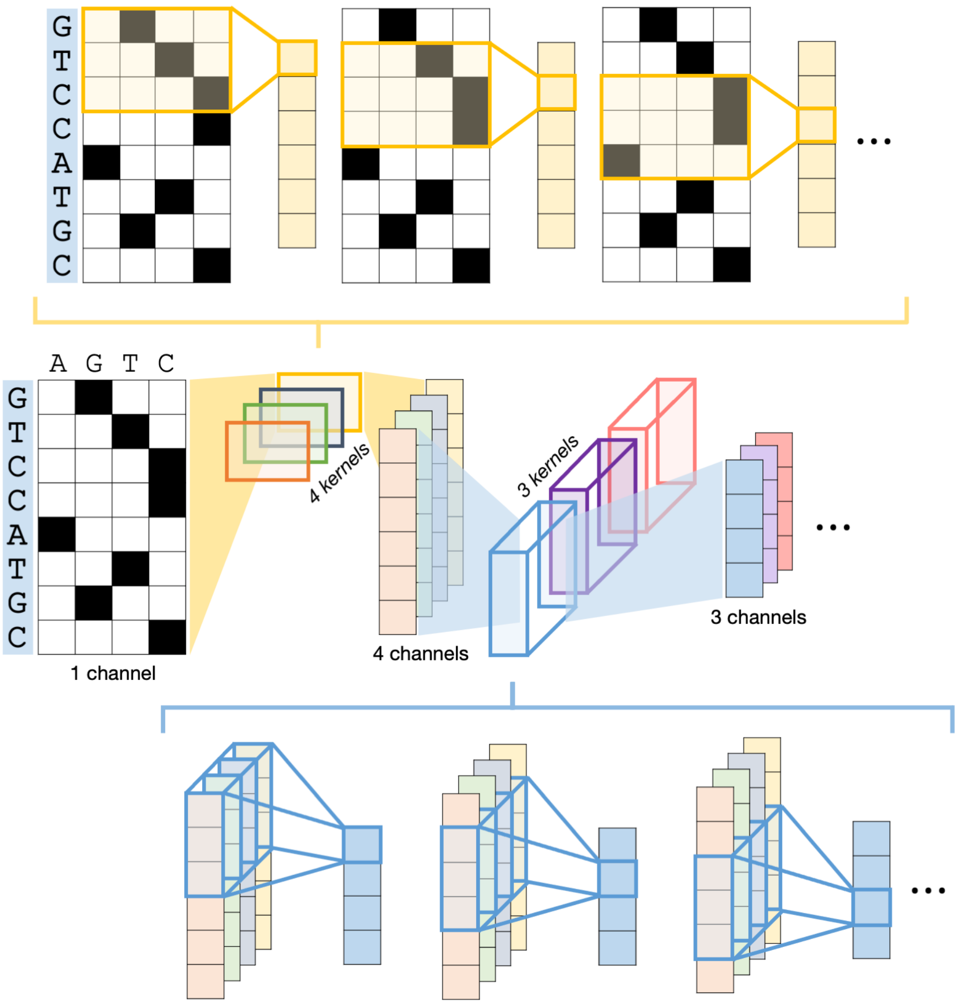


Figure 4.2: **Using Convolutional networks for genetic sequences.** During a convolution, a kernel (or filter) is sled along and across an input image (also called 'channel', e.g. an RGB image has 3 channels), successively examining every residue or nucleotide position in the sequence image. The bottom section shows how a multi channel filter can be applied to multiple input channels at once.

the end or before the start of the sequence image. This means that we can even pad the longest input sequence with extra 0-only rows to account for even longer sequences that might be encountered during testing or clinical deployment. This makes the overall method no longer in need of any alignment algorithm that might otherwise make it unstable.

4.2.2 Encoding interactome information in neural network architecture

The problem I deal with in this subsection is that of incorporating the interactome in the model. While it would be possible to follow Hopf et al. (2014) and concatenate our sequence images and consider the extra-long result as

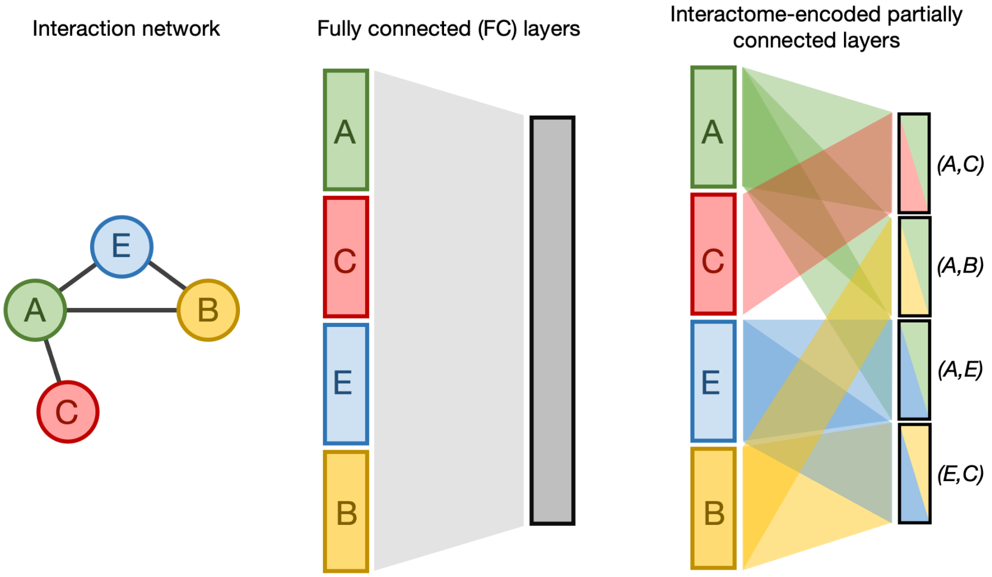

Interaction network
Fully connected (FC) layers
Interactome-encoded partially connected layers
E
A
B
C
A
C
E
B
A
C
E
B
(A,C)
(A,B)
(A,E)
(E,C)

Figure 4.3: **Encoding the interactome into the architecture of a neural network.** Left: the interactome sub-network of interest. Each one of the four components is either an amino acid sequence or a sequence of nucleotides (DNA or RNA). Edges in the graph represent physical interactions. Middle: the conventional fully connected (FC) regime, in which all input nodes are connected to the output ones. Right: the interactome-encoded regime, in which connections are made only when there is a physical interaction between two factors in the network (e.g. when there is an edge between their corresponding nodes).

one sequence, this approach would be inefficient for more than 2 proteins and even more so with sparse networks. Additionally, such an unconstrained, prior-less approach might lead to uninterpretable learning; why would two proteins coevolve if they do not interact with each other? This issue is more obvious with a Potts model than with a VAE because Potts models have an intuitive physical interpretation: what does it mean to have interaction energies for pairs of non-interacting residues? In a neural network and other similar models, such a large number of parameters to train might lead to overfitting wherein information might become 'stored' in neurons and connection weights that have no biological meaning, which would lead to worse generalization.

To address this fundamental challenge, we can enforce a prior on the weights between two layers of neurons such that each pair of interactions in the *gene* network has a corresponding subset of neurons in the second layer of the *neural* network. That way, for each pair of interactions, we draw connections starting from the neurons related to the two interacting proteins to the neurons in the subset that corresponds to that interaction. Such a sensible biological prior allows the neural network to directly use the structure given to it from domain knowledge, preventing uninterpretable learning, overfitting, and poor generalization. (See Figure **4.3** and Figure **4.4** for a simple visual explanation.)

Unsurprisingly, this setup is much more efficient that the fully connected (FC) alternative, and we can quantify the gain in efficiency. Suppose our protein-protein interaction network (e.g. an undirected graph) consists of P proteins of length L, and there are I interactions (edges in the graph). The first layer has PL nodes and the second layer has an arbitrary number of notes, H. In the FC regime, the number of connections is simply PLH. In the encoded interactome regime, we divide the H layer into I subsets of H/I nodes, each subset corresponding to one interaction. For every interaction, we draw LH/I connections twice (once for each protein in the interacting pair). In total, we get $I \cdot 2LH/I = 2LH$, which is $P/2$ times smaller than the FC regime. A network of 10 proteins will have 5 times fewer parameters in the encoded-interactome regime than in the FC one. Of course, we can use these spare parameters to make sure that H/I is large enough for L, which might bring us back to the same number of parameters as in the FC regime, but organized in a much more biologically interpretable way.

4.2.3 A semi-supervised learning framework for variant interpretation

In Section **4.1.3**, I concluded that in order to satisfy the i.i.d assumption (independent and identically distributed), we might need a more stringent requirement than the orthology of the sequences in the training set. I also pinned down the central role of interactome knowledge to guarantee that sequences are subject to identical evolutionary pressure and processes, which satisfies part of the i.i.d. assumption. It is unreasonable to think that epistatis remains unchanged even after the interactome is altered. Therefore, when it comes to learning epistatic mechanisms from sequence data, it is unreasonable to consider a single gene sequence without considering the interactome which contains it.

Because we are interested in human genetic variants, we can only use gene sequences from individual human genomes. Thanks to the method above, we now have a way to encode the small interactome sub-networks in our model (the entire interactome would be way too large). Combining these two insights together, we realize that we can consider an entire sub-network from a human individual as a single input to our models. Where would all this data come from?

Indeed, for any particular disease, there is rarely a wealth of labeled whole genome or exome data large enough to train a deep learning model. Perhaps on the magnitude of a few thousand sequenced genomes per study, wherein each individual genome is accompanied by the phenotypic, clinical, and biochemical indicators relevant to that study. However, genomes from different studies can still be extremely useful, even when they are not accompanied by the phenotypic indicators and values we are interested in. I propose a semi-supervised learning framework to leverage this abundance of unlabeled genomes to improve the accuracy and generalizability of deep learning models that predict the pathogenicity of *variant sub-interactomes*, which we might also call *variant sub-reticulotypes* (from 'reticulum' in Latin, meaning 'network') (Maron et al., 2021).

It is much easier to make the i.i.d. assumption for a diverse set of human sequences than for a diverse set of taxa because of the many aforementioned reasons. Human orthologous sequences trivially have the same function, and so do their interactomes, as they can be assumed to be sampled by a multimodal generative evolutionary process, where each mode could correspond to a population or subpopulation. Therefore, to build a generative model for small human sub-interactomes, one can use the genomes sequenced by any study, regardless of a study's purpose and regardless of a sequence's phenotypic indicators. A list of high-quality databases can be found in the gnomAD database, which is an aggregation of about 80 studies with over 140,000 whole exome and whole genome sequences.

The semi-supervised approach proposed here is operated by two main parts. The first one is the unsupervised generative model that will learn the structure underlying the genome or exome sequence data. By 'underlying structure', we mean the distribution from which evolutionary processes produce their samples. This generative model can be trained using all the sequencing data found in the gnomAD studies, and will ideally learn a useful, information-dense latent space, as well as the encoder to and the decoder from it. This step is akin to clustering the genes of an interactome sub-network. The second part consists of a classifier that takes as input the latent variables found by applying the encoder of the first generative part to the input network, and returns the class prediction we are interested in. The key here is that this classifier, which will be much smaller in size (and therefore complexity) than the generative model, can be trained with the many thousands of labeled data available. And before that, the generative model's parameters can be tuned and optimized with the 140,000 genome or exome sequences (e.g. from the genotype-phenotype studies underlying gnomAD).

This semi-supervised workflow consists of the following steps: (i) choose an interactome sub-network consisting of your genes and proteins of interest; (ii) acquire the corresponding gene sequences from the gnomAD studies and tune the generative model with this large data; (iii) for each individual interactome sub-network in the data set that contains the relevant categories one is interested in predicting (e.g. pathogenic versus benign HCM variants), use the generative model's encoder to encode the interactome sub-network into a vector of latent variables; (iv) use those vectors as labeled training data for the classifier part (with the labels coming from the original labeled sequences). Because the classifier will be smaller, a smaller data set of labeled sequences — which is what is usually available — will suffice.

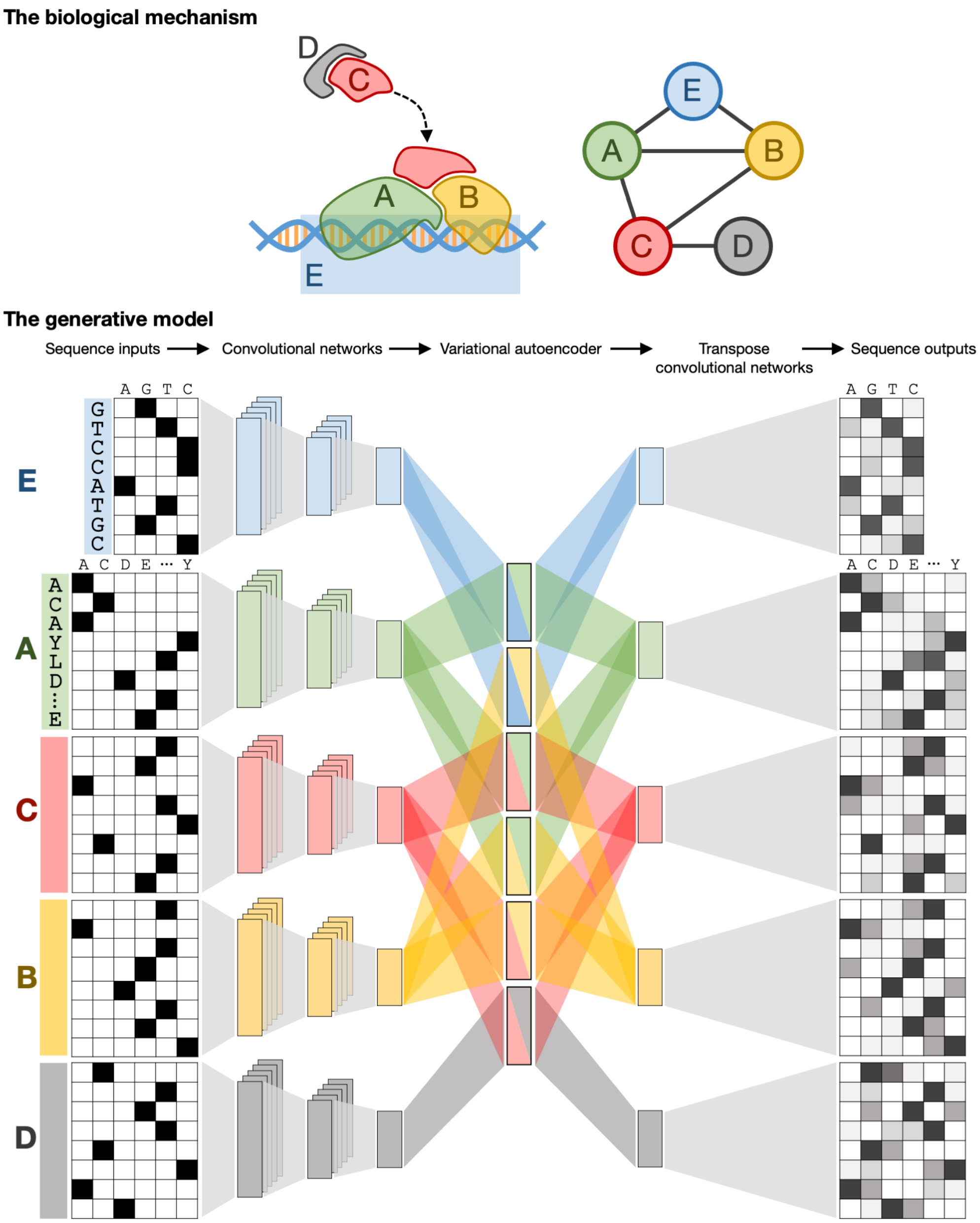


FIGURE 4.4: **An interactome-based VAE model.**

4.3 Three Models

Combining together the proposed solutions above, we derive the generative VAE model shown in Figure **4.4**. First, each sequence in the individual interactome sub-network is transformed into a sequence image, which is then fed to its specific convolutional network described in Section **4.2.1**. This convolutional network outputs a vector of features from the sequence image. The convolutional features from all the sequences in the sub-network are then

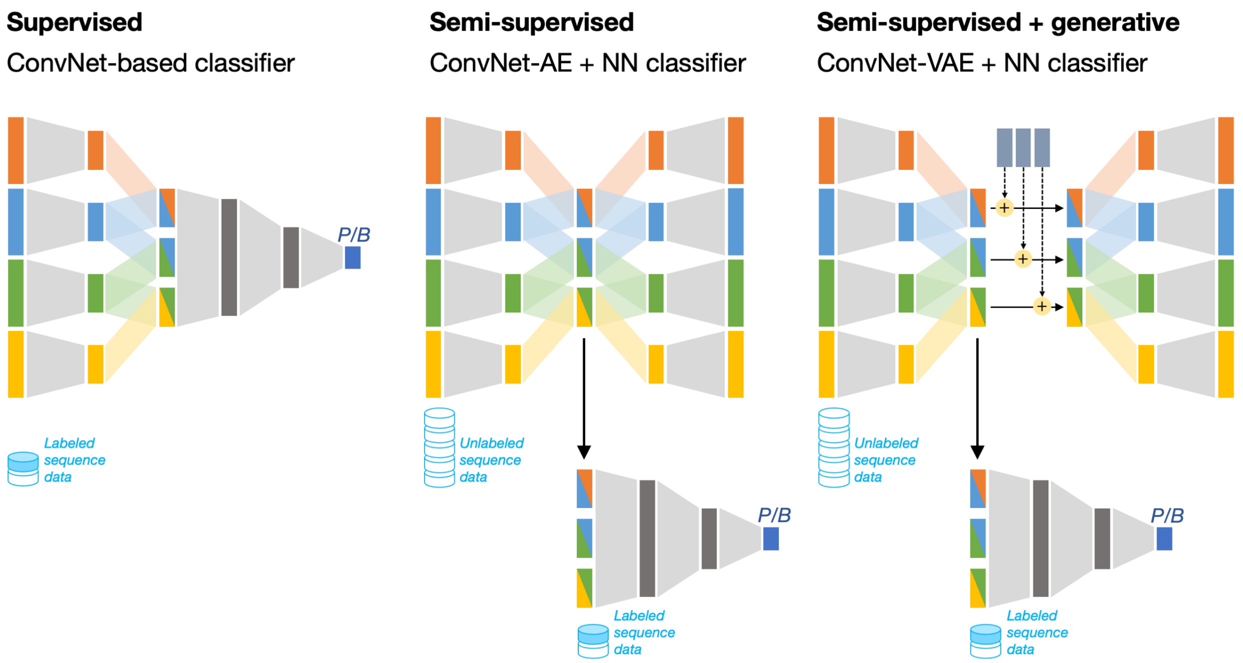


FIGURE 4.5: **Three deep models for the classification of variant interactomes.**

connected to the latent variables nodes according the interactome-encoded partially connected regime described in Section **4.2.2**. After sampling the latent variables, the inputs are reconstructed using 'upsampling' and transpose convolutional networks. Transpose convolutional networks basically do the opposite of what convolutional networks do, starting from a column vector of features, they produce an image. In our case, they reproduce a sequence image that we can turn into an $L \times K$ matrix where each row represents the probability distribution of each nucleotide or amino acid per position. We will call this model 'model 1' throughout the rest of this chapter

While VAEs are great generative models, they can be quite unstable if the variational family or prior distribution over the latent variables is inadequate. To address this potential problem, I propose a second model that replaces the VAE in the first with an autoencoder. Autoencoders are very similar to VAEs in terms of architecture, however they are not probabilistic and the loss only consists of input reconstruction loss (whereas VAE have a loss related to the inferred prior distribution over the latent variables). We will call this model 'model 2' throughout the rest of this chapter. (See the second and third model in Figure **4.5** for a visual explanation of models 1 and 2 respectively.)

Both models 1 and 2 can be trained using the semi-supervised framework elaborated in Section **4.2.3**, as both models can learn a useful latent space in which similar interactome sub-networks cluster together. As we saw earlier, the semi-supervised framework exploits this learned latent structure to better classify variant sub-networks by training a separate classifier on features derived from the latent structure. In the simplest case, the variant sub-networks can be encoded by either model into latent variables, which in turn are fed to the classifier. This classifier does not have to be a neural network, as it can be any model with appropriate sample complexity depending on the labeled data available.

The power of this semi-supervised learning approach is that it can use a diverse set of whole genome or whole exome sequence data (e.g. the 140,000 exomes or genomes from gnomAD) to learn structure that will be present in any labeled data set sampled from the same 'global' distribution. However, there might be cases wherein there is no need to learn a generative distribution; for instance, when there is enough diverse and labeled interactome data to train a classifier or when the network of interest is small, one might as well train the classifier directly. That said, I propose a third model, 'model 3', which solely uses a supervised learning approach. Model 3 shares multiple aspects of its architecture with models 1 and 2, including the sequence images, convolutions networks, and interactome-encoded layer connections. However, it does not reconstruct any inputs because it is fully supervised: the features extracted from the convolutional networks will immediately be forwarded to a classifier that directly predicts the label (e.g. pathogenic or benign). (See the leftmost model in Figure **4.5** for a visual explanation of model 3.)

4.4 Discussion

In this chapter, I highlight three major problems that variant classification models have yet to deal with, and propose one solution to each of them. Finally, I propose three models that combine these solutions and can adapt to different contexts depending on the abundance and diversity of data and the interactome sub-network of interest.

To summarize the dialectic argument spanning the last few sections: for input sequences of variable-length, I propose a convolutional network setup that can accommodate these sequences without requiring gaps and alignments. To integrate the interactome, I propose a method of connecting layers that follows the physical interactions seen in the sub-network of interest. This method encodes into the network meaningful interaction priors that are biologically inspired, thereby preventing uninterpretable learning and overfitting. To resolve the issue of scarce labeled data for a particular disease or phenotype, I propose a semi-supervised approach that tunes the parameters of our generative (or compressing) models using unlabeled genome data. All the models I propose use the first two solutions. Two of them use a semi-supervised framework, of which one is generative (the VAE) and the other is not (the autoencoder).

One major advantage to this approach is that it is *modular*; one can repeat the same process for any interactome sub-network as long as there is labeled data available. An investigator or clinician could choose or design their own sub-network of interest and upload their labeled data from one or more targeted clinical studies. Meanwhile, on the back-end, the models would be assembled, trained using the uploaded labeled data and the unlabeled data from the gnomAD studies, and returned to the investigator or clinician.

These three models are still being implemented, and the data is still being gathered across the 80 gnomAD studies. However, even in the case they reach high and clinically acceptable performance levels, there remains limitations to solve. For instance, unless the convolutional neural networks are *fully* convolutional, there will always be an upper bound on the length of an input sequence. In addition, these methods are very dependent on our knowledge of the human interactome, which is presently very incomplete. False interactions or missed interactions might lead to overfitting and poor generalization.

5
Conclusion

Whole genome and exome sequencing are becoming accessible and instrumental tools for the diagnosis and treatment of Mendelian disorders. Their soaring use has led to the discovery of hundreds of millions of genetic variants, the majority of which were recently seen for the very first time, and almost all of which are of unknown significance. The need to interpret all of these genetic variants and discern the pathogenic ones from the benign ones has spurred considerable efforts to develop computational methods that address this challenging task at scale.

The purpose of this thesis is to contribute to the ensemble of pathogenicity prediction methods by introducing deep learning models that strive to resolve some of the main problems that methods in the field have had to deal with. To do this, I first reviewed the most common approaches used thus far, developing a roadmap of methods defined along data features, model complexity, and encoded domain knowledge. This roadmap of methods helps us better understand the balance between data and model complexity in existing methods, as well as their weaknesses and strengths. The insights made from this analysis allowed us to introduce new generalizable deep learning methods for variant pathogenicity classification that leveraged the interactome, the totality of molecular interactions in the cell. Along the way, I reproduced and extended a well-established baseline approach for variant classification, creating a publicly available codebase. My hope is that the approaches introduced in this thesis will lead to improved clinical-grade variant prioritization approaches as well as a framework to extract robust biological insights from first principles.

References

Adzhubei, I. A., Schmidt, S., Peshkin, L., Ramensky, V. E., Gerasimova, A., Bork, P., Kondrashov, A. S., and Sunyaev, S. R. (2010). A method and server for predicting damaging missense mutations. *Nature Methods*, 7(4):248–249. Number: 4 Publisher: Nature Publishing Group.

Aken, B. L., Achuthan, P., Akanni, W., Amode, M. R., Bernsdorff, F., Bhai, J., Billis, K., Carvalho-Silva, D., Cummins, C., Clapham, P., Gil, L., Girón, C. G., Gordon, L., Hourlier, T., Hunt, S. E., Janacek, S. H., Juettemann, T., Keenan, S., Laird, M. R., Lavidas, I., Maurel, T., McLaren, W., Moore, B., Murphy, D. N., Nag, R., Newman, V., Nuhn, M., Ong, C. K., Parker, A., Patricio, M., Riat, H. S., Sheppard, D., Sparrow, H., Taylor, K., Thormann, A., Vullo, A., Walts, B., Wilder, S. P., Zadissa, A., Kostadima, M., Martin, F. J., Muffato, M., Perry, E., Ruffier, M., Staines, D. M., Trevanion, S. J., Cunningham, F., Yates, A., Zerbino, D. R., and Flicek, P. (2017). Ensembl 2017. *Nucleic Acids Research*, 45(D1):D635–D642.

Amendola, L. M., Jarvik, G. P., Leo, M. C., McLaughlin, H. M., Akkari, Y., Amaral, M. D., Berg, J. S., Biswas, S., Bowling, K. M., Conlin, L. K., Cooper, G. M., Dorschner, M. O., Dulik, M. C., Ghazani, A. A., Ghosh, R., Green, R. C., Hart, R., Horton, C., Johnston, J. J., Lebo, M. S., Milosavljevic, A., Ou, J., Pak, C. M., Patel, R. Y., Punj, S., Richards, C. S., Salama, J., Strande, N. T., Yang, Y., Plon, S. E., Biesecker, L. G., and Rehm, H. L. (2016). Performance of ACMG-AMP Variant-Interpretation Guidelines among Nine Laboratories in the Clinical Sequencing Exploratory Research Consortium. *The American Journal of Human Genetics*, 99(1):247. Publisher: Elsevier.

Anderson, D. W., McKeown, A. N., and Thornton, J. W. (2015). Intermolecular epistasis shaped the function and evolution of an ancient transcription factor and its DNA binding sites. *eLife*, 4:e07864. Publisher: eLife Sciences Publications, Ltd.

Balakrishnan, S., Kamisetty, H., Carbonell, J. G., Lee, S.-I., and Langmead, C. J. (2011). Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.22934.

Barabási, A.-L., Gulbahce, N., and Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68. Number: 1 Publisher: Nature Publishing Group.

Bendixsen, D. P., Østman, B., and Hayden, E. J. (2017). Negative Epistasis in Experimental RNA Fitness Landscapes. *Journal of Molecular Evolution*, 85(5):159–168.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877. arXiv: 1601.00670.

Breen, M. S., Kemena, C., Vlasov, P. K., Notredame, C., and Kondrashov, F. A. (2012). Epistasis as the primary factor in molecular evolution. *Nature*, 490(7421):535–538. Number: 7421 Publisher: Nature Publishing Group.

Cargill, M., Altshuler, D., Ireland, J., Sklar, P., Ardlie, K., Patil, N., Lane, C. R., Lim, E. P., Kalyanaraman, N., Nemesh, J., Ziaugra, L., Friedland, L., Rolfe, A., Warrington, J., Lipshutz, R., Daley, G. Q., and Lander, E. S. (1999). Characterization of single-nucleotide polymorphisms in coding regions of human genes. *Nature Genetics*, 22(3):231–238. Number: 3 Publisher: Nature Publishing Group.

Choi, Y., Sims, G. E., Murphy, S., Miller, J. R., and Chan, A. P. (2012). Predicting the Functional Effect of Amino Acid Substitutions and Indels. *PLOS ONE*, 7(10):e46688. Publisher: Public Library of Science.

Chong, J. X., Buckingham, K. J., Jhangiani, S. N., Boehm, C., Sobreira, N., Smith, J. D., Harrell, T. M., McMillin, M. J., Wiszniewski, W., Gambin, T., Coban Akdemir, Z. H., Doheny, K., Scott, A. F., Avramopoulos, D., Chakravarti, A., Hoover-Fong, J., Mathews, D., Witmer, P. D., Ling, H., Hetrick, K., Watkins, L., Patterson, K. E., Reinier, F., Blue, E., Muzny, D., Kircher, M., Bilguvar, K., López-Giráldez, F., Sutton, V. R., Tabor, H. K., Leal, S. M., Gunel, M., Mane, S., Gibbs, R. A., Boerwinkle, E., Hamosh, A., Shendure, J., Lupski, J. R., Lifton, R. P., Valle, D., Nickerson, D. A., and Bamshad, M. J. (2015). The Genetic Basis of Mendelian Phenotypes: Discoveries, Challenges, and Opportunities. *The American Journal of Human Genetics*, 97(2):199–215. Publisher: Elsevier.

Ding, X., Zou, Z., and Brooks Iii, C. L. (2019). Deciphering protein evolution and fitness landscapes with latent space models. *Nature Communications*, 10(1):5644. Number: 1 Publisher: Nature Publishing Group.

Doersch, C. (2021). Tutorial on Variational Autoencoders. *arXiv:1606.05908 [cs, stat]*. arXiv: 1606.05908.

Doig, K. D., Fellowes, A., Bell, A. H., Seleznev, A., Ma, D., Ellul, J., Li, J., Doyle, M. A., Thompson, E. R., Kumar, A., Lara, L., Vedururu, R., Reid, G., Conway, T., Papenfuss, A. T., and Fox, S. B. (2017). PathOS: a decision support system for reporting high throughput sequencing of cancers in clinical diagnostic laboratories. *Genome Medicine*, 9(1):38.

Dunham, I., Kundaje, A., Aldred, S. F., Collins, P. J., Davis, C. A., Doyle, F., Epstein, C. B., Frietze, S., Harrow, J., Kaul, R., Khatun, J., Lajoie, B. R., Landt, S. G., Lee, B.-K., Pauli, F., Rosenbloom, K. R., Sabo, P., Safi, A., Sanyal, A., Shoresh, N., Simon, J. M., Song, L., Trinklein, N. D., Altshuler, R. C., Birney, E., Brown, J. B., Cheng, C., Djebali, S., Dong, X., Dunham, I., Ernst, J., Furey, T. S., Gerstein, M., Giardine, B., Greven, M., Hardison, R. C., Harris, R. S., Herrero, J., Hoffman, M. M., Iyer, S., Kellis, M., Khatun, J., Kheradpour, P., Kundaje, A., Lassmann, T., Li, Q., Lin, X., Marinov, G. K., Merkel, A., Mortazavi, A., Parker, S. C. J., Reddy, T. E., Rozowsky, J., Schlesinger, F., Thurman, R. E., Wang, J., Ward, L. D., Whitfield, T. W., Wilder, S. P., Wu, W., Xi, H. S., Yip, K. Y., Zhuang, J., Bernstein, B. E., Birney, E., Dunham, I., Green, E. D., Gunter, C., Snyder, M., Pazin, M. J., Lowdon, R. F., Dillon, L. A. L., Adams, L. B., Kelly, C. J., Zhang, J., Wexler, J. R., Green, E. D., Good, P. J., Feingold, E. A., Bernstein, B. E., Birney, E., Crawford, G. E., Dekker, J., Elnitski, L., Farnham, P. J., Gerstein, M., Giddings, M. C., Gingeras, T. R., Green, E. D., Guigó, R., Hardison, R. C., Hubbard, T. J., Kellis, M., Kent, W. J., Lieb, J. D., Margulies, E. H., Myers, R. M., Snyder, M., Stamatoyannopoulos, J. A., Tenenbaum, S. A., Weng, Z., White, K. P., Wold, B., Khatun, J., Yu, Y., Wrobel, J., Risk, B. A., Gunawardena, H. P., Kuiper, H. C., Maier, C. W., Xie, L., Chen, X., Giddings, M. C., Bernstein, B. E., Epstein, C. B., Shoresh, N., Ernst, J., Kheradpour, P., Mikkelsen, T. S., Gillespie, S., Goren, A., Ram, O., Zhang, X., Wang, L., Issner, R., Coyne, M. J., Durham, T., Ku, M., Truong, T., Ward, L. D., Altshuler, R. C., Eaton, M. L., Kellis, M., Djebali, S., Davis, C. A., Merkel, A., Dobin, A., Lassmann, T., Mortazavi, A., Tanzer, A., Lagarde, J., Lin, W., Schlesinger, F., Xue, C., Marinov, G. K., Khatun, J., Williams, B. A., Zaleski, C., Rozowsky, J., Röder, M., Kokocinski, F., Abdelhamid, R. F., Alioto, T., Antoshechkin, I., Baer, M. T., Batut, P., Bell, I., Bell, K., Chakrabortty, S., Chen, X., Chrast, J., Curado, J., Derrien, T., Drenkow, J., Dumais, E., Dumais, J., Duttagupta, R., Fastuca, M., Fejes-Toth, K., Ferreira, P., Foissac, S., Fullwood, M. J., Gao, H., Gonzalez, D., Gordon, A., Gunawardena, H. P., Howald, C., Jha, S., Johnson, R., Kapranov, P., King, B., Kingswood, C., Li, G., Luo, O. J., Park, E., Preall, J. B., Presaud, K., Ribeca, P., Risk, B. A., Robyr, D., Ruan, X., Sammeth, M., Sandhu, K. S., Schaeffer, L., See, L.-H., Shahab, A., Skancke, J., Suzuki,

A. M., Takahashi, H., Tilgner, H., Trout, D., Walters, N., Wang, H., Wrobel, J., Yu, Y., Hayashizaki, Y., Harrow, J., Gerstein, M., Hubbard, T. J., Reymond, A., Antonarakis, S. E., Hannon, G. J., Giddings, M. C., Ruan, Y., Wold, B., Carninci, P., Guigó, R., Gingeras, T. R., Rosenbloom, K. R., Sloan, C. A., Learned, K., Malladi, V. S., Wong, M. C., Barber, G. P., Cline, M. S., Dreszer, T. R., Heitner, S. G., Karolchik, D., Kent, W. J., Kirkup, V. M., Meyer, L. R., Long, J. C., Maddren, M., Raney, B. J., Furey, T. S., Song, L., Grasfeder, L. L., Giresi, P. G., Lee, B.-K., Battenhouse, A., Sheffield, N. C., Simon, J. M., Showers, K. A., Safi, A., London, D., Bhinge, A. A., Shestak, C., Schaner, M. R., Ki Kim, S., Zhang, Z. Z., Mieczkowski, P. A., Mieczkowska, J. O., Liu, Z., McDaniell, R. M., Ni, Y., Rashid, N. U., Kim, M. J., Adar, S., Zhang, Z., Wang, T., Winter, D., Keefe, D., Birney, E., Iyer, V. R., Lieb, J. D., Crawford, G. E., Li, G., Sandhu, K. S., Zheng, M., Wang, P., Luo, O. J., Shahab, A., Fullwood, M. J., Ruan, X., Ruan, Y., Myers, R. M., Pauli, F., Williams, B. A., Gertz, J., Marinov, G. K., Reddy, T. E., Vielmetter, J., Partridge, E., Trout, D., Varley, K. E., Gasper, C., The ENCODE Project Consortium, Overall coordination (data analysis coordination), Data production leads (data production), Lead analysts (data analysis), Writing group, NHGRI project management (scientific management), Principal investigators (steering committee), Boise State University and University of North Carolina at Chapel Hill Proteomics groups (data production and analysis), Broad Institute Group (data production and analysis), Cold Spring Harbor, Center for Genomic Regulation, B. R. S. I. U. o. L. G. I. o. S. g. d. p. a. a. U. o. G., Data coordination center at UC Santa Cruz (production data coordination), Duke University, University of Texas, A. U. o. N. C.-C. H. g. d. p. a. a. E., Genome Institute of Singapore group (data production and analysis), and HudsonAlpha Institute, UC Irvine, S. g. d. p. a. a.-C. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74. Number: 7414 Publisher: Nature Publishing Group.

Eilbeck, K., Quinlan, A., and Yandell, M. (2017). Settling the score: variant prioritization and Mendelian disease. *Nature Reviews Genetics*, 18(10):599–612. Number: 10 Publisher: Nature Publishing Group.

Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013). Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E*, 87(1):012707. arXiv: 1211.1281.

Felsenstein, J. (1981). Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376.

Gulko, B., Hubisz, M. J., Gronau, I., and Siepel, A. (2015). A method for calculating probabilities of fitness consequences for point mutations across the human genome. *Nature Genetics*, 47(3):276–283. Number: 3 Publisher: Nature Publishing Group.

Hart, G. T., Ramani, A. K., and Marcotte, E. M. (2006). How complete are current yeast and human protein-interaction networks? *Genome Biology*, 7(11):120.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915–10919.

Henikoff, S. and Henikoff, J. G. (1994). Position-based sequence weights. *Journal of Molecular Biology*, 243(4):574–578.

Hopf, T., Colwell, L., Sheridan, R., Rost, B., Sander, C., and Marks, D. (2012). Three-Dimensional Structures of Membrane Proteins from Genomic Sequencing. *Cell*, 149(7):1607–1621.

Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P. I., Springer, M., Sander, C., and Marks, D. S. (2017). Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135. Number: 2 Publisher: Nature Publishing Group.

Hopf, T. A., Schärfe, C. P. I., Rodrigues, J. P. G. L. M., Green, A. G., Kohlbacher, O., Sander, C., Bonvin, A. M. J. J., and Marks, D. S. (2014). Sequence co-evolution gives 3D contacts and structures of protein complexes. *eLife*, 3:e03430. Publisher: eLife Sciences Publications, Ltd.

Hu, H., Huff, C. D., Moore, B., Flygare, S., Reese, M. G., and Yandell, M. (2013). VAAST 2.0: Improved Variant Classification and Disease-Gene Identification Using a Conservation-Controlled Amino Acid Substitution Matrix. *Genetic Epidemiology*, 37(6):622–634.

Hu, J. and Ng, P. C. (2012). Predicting the effects of frameshifting indels. *Genome Biology*, 13(2):R9.

Jordan, D. M., Kiezun, A., Baxter, S. M., Agarwala, V., Green, R. C., Murray, M. F., Pugh, T., Lebo, M. S., Rehm, H. L., Funke, B. H., and Sunyaev, S. R. (2011). Development and Validation of a Computational Method for Assessment of Missense Variants in Hypertrophic Cardiomyopathy. *The American Journal of Human Genetics*, 88(2):183–192. Publisher: Elsevier.

Jukes, T. H. and Cantor, C. R. (1969). CHAPTER 24 - Evolution of Protein Molecules. In Munro, H. N., editor, *Mammalian Protein Metabolism*, pages 21–132. Academic Press.

Karczewski, K. J., Francioli, L. C., Tiao, G., Cummings, B. B., Alföldi, J., Wang, Q., Collins, R. L., Laricchia, K. M., Ganna, A., Birnbaum, D. P., Gauthier, L. D., Brand, H., Solomonson, M., Watts, N. A., Rhodes, D., Singer-Berk, M., England, E. M., Seaby, E. G., Kosmicki, J. A., Walters, R. K., Tashman, K., Farjoun, Y., Banks, E., Poterba, T., Wang, A., Seed, C., Whiffin, N., Chong, J. X., Samocha, K. E., Pierce-Hoffman, E., Zappala, Z., O'Donnell-Luria, A. H., Minikel, E. V., Weisburd, B., Lek, M., Ware, J. S., Vittal, C., Armean, I. M., Bergelson, L., Cibulskis, K., Connolly, K. M., Covarrubias, M., Donnelly, S., Ferriera, S., Gabriel, S., Gentry, J., Gupta, N., Jeandet, T., Kaplan, D., Llanwarne, C., Munshi, R., Novod, S., Petrillo, N., Roazen, D., Ruano-Rubio, V., Saltzman, A., Schleicher, M., Soto, J., Tibbetts, K., Tolonen, C., Wade, G., Talkowski, M. E., Neale, B. M., Daly, M. J., and MacArthur, D. G. (2020). The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature*, 581(7809):434–443. Number: 7809 Publisher: Nature Publishing Group.

Kingma, D. P. and Welling, M. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392. arXiv: 1906.02691.

Kircher, M., Witten, D. M., Jain, P., O'Roak, B. J., Cooper, G. M., and Shendure, J. (2014). A general framework for estimating the relative pathogenicity of human genetic variants. *Nature Genetics*, 46(3):310–315. Number: 3 Publisher: Nature Publishing Group.

Lek, M., Karczewski, K. J., Minikel, E. V., Samocha, K. E., Banks, E., Fennell, T., O'Donnell-Luria, A. H., Ware, J. S., Hill, A. J., Cummings, B. B., Tukiainen, T., Birnbaum, D. P., Kosmicki, J. A., Duncan, L. E., Estrada, K., Zhao, F., Zou, J., Pierce-Hoffman, E., Berghout, J., Cooper, D. N., Deflaux, N., DePristo, M., Do, R., Flannick, J., Fromer, M., Gauthier, L., Goldstein, J., Gupta, N., Howrigan, D., Kiezun, A., Kurki, M. I., Moonshine, A. L., Natarajan, P., Orozco, L., Peloso, G. M., Poplin, R., Rivas, M. A., Ruano-Rubio, V., Rose, S. A., Ruderfer, D. M., Shakir, K., Stenson, P. D., Stevens, C., Thomas, B. P., Tiao, G., Tusie-Luna, M. T., Weisburd, B., Won, H.-H., Yu, D., Altshuler, D. M., Ardissino, D., Boehnke, M., Danesh, J., Donnelly, S., Elosua, R., Florez, J. C., Gabriel, S. B., Getz, G., Glatt, S. J., Hultman, C. M., Kathiresan, S., Laakso, M., McCarroll, S., McCarthy, M. I., McGovern, D., McPherson, R., Neale, B. M., Palotie, A., Purcell, S. M., Saleheen, D., Scharf, J. M., Sklar, P., Sullivan, P. F., Tuomilehto, J., Tsuang, M. T., Watkins, H. C., Wilson, J. G., Daly, M. J., and MacArthur, D. G. (2016). Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, 536(7616):285–291. Number: 7616 Publisher: Nature Publishing Group.

Lupas, A., Dyke, M. V., and Stock, J. (1991). Predicting coiled coils from protein sequences. *Science*, 252(5009):1162–1164. Publisher: American Association for the Advancement of Science Section: Reports.

Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C. (2011). Protein 3D Structure Computed from Evolutionary Sequence Variation. *PLOS ONE*, 6(12):e28766. Publisher: Public Library of Science.

Maron, B. A., Wang, R.-S., Shevtsov, S., Drakos, S. G., Arons, E., Wever-Pinzon, O., Huggins, G. S., Samokhin, A. O., Oldham, W. M., Aguib, Y., Yacoub, M. H., Rowin, E. J., Maron, B. J., Maron, M. S., and Loscalzo, J. (2021). Individualized interactomes for network-based precision medicine in hypertrophic cardiomyopathy with implications for other clinical pathophenotypes. *Nature Communications*, 12(1):873. Number: 1 Publisher: Nature Publishing Group.

McCandlish, D. M., Shah, P., and Plotkin, J. B. (2016). Epistasis and the Dynamics of Reversion in Molecular Evolution. *Genetics*, 203(3):1335–1351. Publisher: Genetics Section: Investigations.

McVean, G. A., Altshuler (Co-Chair), D. M., Durbin (Co-Chair), R. M., Abecasis, G. R., Bentley, D. R., Chakravarti, A., Clark, A. G., Donnelly, P., Eichler, E. E., Flicek, P., Gabriel, S. B., Gibbs, R. A., Green, E. D., Hurles, M. E., Knoppers, B. M., Korbel, J. O., Lander, E. S., Lee, C., Lehrach, H., Mardis, E. R., Marth, G. T., McVean, G. A., Nickerson, D. A., Schmidt, J. P., Sherry, S. T., Wang, J., Wilson, R. K., Gibbs (Principal Investigator), R. A., Dinh, H., Kovar, C., Lee, S., Lewis, L., Muzny, D., Reid, J., Wang, M., Wang (Principal Investigator), J., Fang, X., Guo, X., Jian, M., Jiang, H., Jin, X., Li, G., Li, J., Li, Y., Li, Z., Liu, X., Lu, Y., Ma, X., Su, Z., Tai, S., Tang, M., Wang, B., Wang, G., Wu, H., Wu, R., Yin, Y., Zhang, W., Zhao, J., Zhao, M., Zheng, X., Zhou, Y., Lander (Principal Investigator), E. S., Altshuler, D. M., Gabriel (Co-Chair), S. B., Gupta, N., Flicek (Principal Investigator), P., Clarke, L., Leinonen, R., Smith, R. E., Zheng-Bradley, X., Bentley (Principal Investigator), D. R., Grocock, R., Humphray, S., James, T., Kingsbury, Z., Lehrach (Principal Investigator), H., Sudbrak (Project Leader), R., Albrecht, M. W., Amstislavskiy, V. S., Borodina, T. A., Lienhard, M., Mertes, F., Sultan, M., Timmermann, B., Yaspo, M.-L., Sherry (Principal Investigator), S. T., McVean (Principal Investigator), G. A., Mardis (Co-Principal Investigator) (Co-Chair), E. R., Wilson (Co-Principal Investigator), R. K., Fulton, L., Fulton, R., Weinstock, G. M., Durbin (Principal Investigator), R. M., Balasubramaniam, S., Burton, J., Danecek, P., Keane, T. M., Kolb-Kokocinski, A., McCarthy, S., Stalker, J., Quail, M., Schmidt (Principal Investigator), J. P., Davies, C. J., Gollub, J., Webster, T., Wong, B., Zhan, Y., Auton (Principal Investigator), A., Gibbs (Principal Investigator), R. A., Yu (Project Leader), F., Bainbridge, M., Challis, D., Evani, U. S., Lu, J., Muzny, D., Nagaswamy, U., Reid, J., Sabo, A., Wang, Y., Yu, J., Wang (Principal Investigator), J., Coin, L. J. M., Fang, L., Guo, X., Jin, X., Li, G., Li, Q., Li, Y., Li, Z., Lin, H., Liu, B., Luo, R., Qin, N., Shao, H., Wang, B., Xie, Y., Ye, C., Yu, C., Zhang, F., Zheng, H., Zhu, H., Marth (Principal Investigator), G. T., Garrison, E. P., Kural, D., Lee, W.-P., Fung Leong, W., Ward, A. N., Wu, J., Zhang, M., Lee (Principal Investigator), C., Griffin, L., Hsieh, C.-H., Mills, R. E., Shi, X., von Grotthuss, M., Zhang, C., Daly (Principal Investigator), M. J., DePristo (Project Leader), M. A., Altshuler, D. M., Banks, E., Bhatia, G., Carneiro, M. O., del Angel, G., Gabriel, S. B., Genovese, G., Gupta, N., Handsaker, R. E., Hartl, C., Lander, E. S., McCarroll, S. A., Nemesh, J. C., Poplin, R. E., Schaffner, S. F., Shakir, K., Yoon (Principal Investigator), S. C., Lihm, J., Makarov, V., Jin (Principal Investigator), H., Kim, W., Cheol Kim, K., Korbel (Principal Investigator), J. O., Rausch, T., Flicek (Principal Investigator), P., Beal, K., Clarke, L., Cunningham, F., Herrero, J., McLaren, W. M., Ritchie, G. R. S., Smith, R. E., Zheng-Bradley, X., Clark (Principal Investigator), A. G., Gottipati, S., Keinan, A., Rodriguez-Flores, J. L., Sabeti (Principal Investigator), P. C., Grossman, S. R., Tabrizi, S., Tariyal, R., Cooper (Principal Investigator), D. N., Ball, E. V., Stenson, P. D., Bentley (Principal Investigator), D. R., Barnes, B., Bauer, M., Keira Cheetham, R., Cox, T., Eberle, M., Humphray, S., Kahn, S., Murray, L., Peden, J., Shaw, R., Ye (Principal Investigator), K., Batzer (Principal Investigator), M. A., Konkel, M. K., Walker, J. A., MacArthur (Principal Investigator), D. G., Lek, M., Sudbrak (Project Leader), Amstislavskiy, V. S., Herwig, R., Shriver (Principal Investigator), M. D., Bustamante (Principal Investigator), C. D., Byrnes, J. K., De La Vega, F. M., Gravel, S., Kenny, E. E., Kidd, J. M., Lacroute, P., Maples, B. K., Moreno-Estrada, A., Zakharia, F., Halperin (Principal Investigator), E., Baran, Y., Craig (Principal Investigator), D. W., Christoforides, A., Homer, N., Izatt, T., Kurdoglu, A. A., Sinari, S. A., Squire, K., Sherry (Principal Investigator), S. T., Xiao, C., Sebat (Principal Investigator), J., Bafna, V., Ye, K., Burchard (Principal Investigator), E. G., Hernandez (Principal Investigator), R. D., Gignoux, C. R., Haussler (Principal Investigator), D., Katzman, S. J., James Kent, W., Howie, B., Ruiz-Linares (Principal Investigator), A., The 1000 Genomes Project Consortium, Corresponding Author, Steering committee, Production group:, Baylor College of Medicine, BGI-Shenzhen, Broad Institute of MIT and Harvard, European Bioinformatics Institute, Illumina, Max Planck Institute for Molecular Genetics, US National Institutes of Health, University of Oxford, Washington University in St Louis, Wellcome Trust Sanger Institute, Analysis group:, Affymetrix, Albert Einstein College of Medicine, Boston College, Brigham and Women's Hospital, Cold Spring Harbor Laboratory, Dankook University, European Molecular Biology Laboratory, Cornell University, Harvard University, Human Gene Mutation Database, Leiden University Medical Center, Louisiana State University, Massachusetts General Hospital, Pennsylvania State University, Stanford University, Tel-Aviv University, Translational Genomics Research Institute, University of California, S. D., University of California, S. F., University of California, S. C., University of Chicago, University College London, and University of Geneva (2012). An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65. Number: 7422 Publisher: Nature Publishing Group.

Menche, J., Sharma, A., Kitsak, M., Ghiassian, S. D., Vidal, M., Loscalzo, J., and Barabási, A.-L. (2015). Uncovering disease-disease relationships through the incomplete interactome. *Science*, 347(6224). Publisher: American Association for the Advancement of Science Section: Research Article.

Meyer, L. R., Zweig, A. S., Hinrichs, A. S., Karolchik, D., Kuhn, R. M., Wong, M., Sloan, C. A., Rosenbloom, K. R., Roe, G., Rhead, B., Raney, B. J., Pohl, A., Malladi, V. S., Li, C. H., Lee, B. T., Learned, K., Kirkup, V., Hsu, F., Heitner, S., Harte, R. A., Haeussler, M., Guruvadoo, L., Goldman, M., Giardine, B. M., Fujita, P. A., Dreszer, T. R., Diekhans, M., Cline, M. S., Clawson, H., Barber, G. P., Haussler, D., and Kent, W. J. (2013). The UCSC Genome Browser database: extensions and updates 2013. *Nucleic Acids Research*, 41(D1):D64–D69. Publisher: Oxford Academic.

Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301. Publisher: National Academy of Sciences Section: PNAS Plus.

Ng, P. C. and Henikoff, S. (2006). Predicting the effects of amino acid substitutions on protein function. *Annual Review of Genomics and Human Genetics*, 7:61–80.

Ovchinnikov, S., Kamisetty, H., and Baker, D. (2014). Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information. *eLife*, 3:e02030. Publisher: eLife Sciences Publications, Ltd.

Ovchinnikov, S., Kinch, L., Park, H., Liao, Y., Pei, J., Kim, D. E., Kamisetty, H., Grishin, N. V., and Baker, D. (2015). Large-scale determination of previously unsolved protein structures using evolutionary information. *eLife*, 4:e09248. Publisher: eLife Sciences Publications, Ltd.

Petrovski, S., Wang, Q., Heinzen, E. L., Allen, A. S., and Goldstein, D. B. (2013). Genic Intolerance to Functional Variation and the Interpretation of Personal Genomes. *PLOS Genetics*, 9(8):e1003709. Publisher: Public Library of Science.

Richards, S., Aziz, N., Bale, S., Bick, D., Das, S., Gastier-Foster, J., Grody, W. W., Hegde, M., Lyon, E., Spector, E., Voelkerding, K., and Rehm, H. L. (2015). Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. *Genetics in Medicine*, 17(5):405–423. Number: 5 Publisher: Nature Publishing Group.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. (2018). Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822. Number: 10 Publisher: Nature Publishing Group.

Shah, N., Hou, Y.-C. C., Yu, H.-C., Sainger, R., Dec, E., Perkins, B., Caskey, C. T., Venter, J. C., and Telenti, A. (2016). Identification of misclassified ClinVar variants using disease population prevalence. *bioRxiv*, page 075416. Publisher: Cold Spring Harbor Laboratory Section: New Results.

Stumpf, M. P. H., Thorne, T., de Silva, E., Stewart, R., An, H. J., Lappe, M., and Wiuf, C. (2008). Estimating the size of the human interactome. *Proceedings of the National Academy of Sciences*, 105(19):6959–6964.

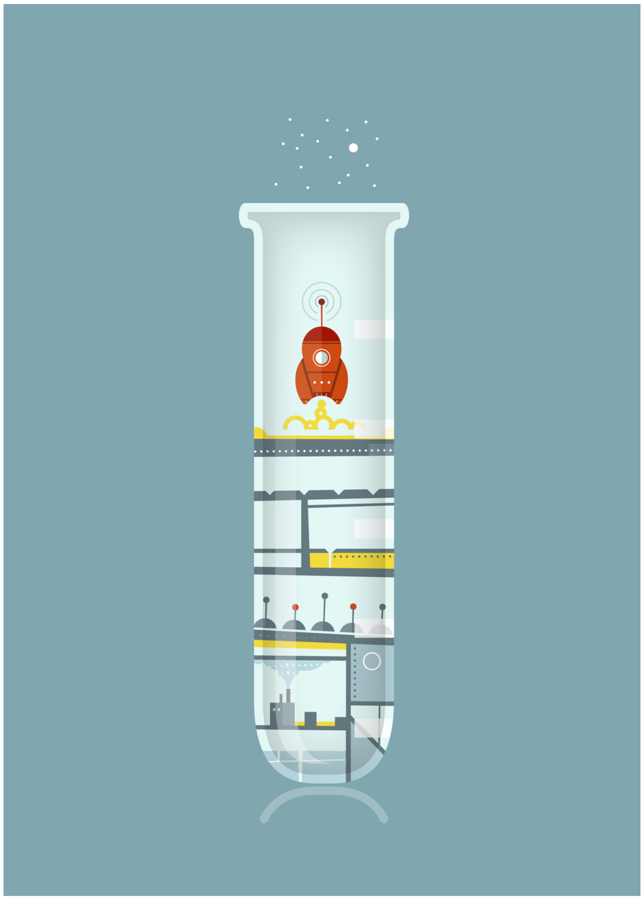
Weinreb, C., Riesselman, A., Ingraham, J., Gross, T., Sander, C., and Marks, D. (2016). 3D RNA and Functional Interactions from Evolutionary Couplings. *Cell*, 165(4):963–975.

Weinreich, D. M., Lan, Y., Wylie, C. S., and Heckendorn, R. B. (2013). Should evolutionary geneticists worry about higher-order epistasis? *Current Opinion in Genetics & Development*, 23(6):700–707.

Whiffin, N., Minikel, E., Walsh, R., O'Donnell-Luria, A. H., Karczewski, K., Ing, A. Y., Barton, P. J. R., Funke, B., Cook, S. A., MacArthur, D., and Ware, J. S. (2017). Using high-resolution variant frequencies to empower clinical genome interpretation. *Genetics in Medicine*, 19(10):1151–1158. Number: 10 Publisher: Nature Publishing Group.

Wioland, H., Woodhouse, F. G., Dunkel, J., and Goldstein, R. E. (2016). Ferromagnetic and antiferromagnetic order in bacterial vortex lattices. *Nature Physics*, 12(4):341–345. arXiv: 1511.05000.

Yang, G., Anderson, D. W., Baier, F., Dohmen, E., Hong, N., Carr, P. D., Kamerlin, S. C. L., Jackson, C. J., Bornberg-Bauer, E., and Tokuriki, N. (2019). Higher-order epistasis shapes the fitness landscape of a xenobiotic-degrading enzyme. *Nature Chemical Biology*, 15(11):1120–1128. Number: 11 Publisher: Nature Publishing Group.



This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. The above illustration, "Science Experiment 02", was created by Ben Schlitter and released under CC BY-NC-ND 3.0. A template that can be used to format a PhD thesis with this look and feel has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/Dissertate or from its author, Jordan Suchow, at suchow@post.harvard.edu.