Predicting expression levels of *de novo* protein designs in yeast through machine learning.

Patrick Slogic

A thesis

submitted in partial fulfilment of the

requirements for the degree of:

Master of Science in Bioengineering

University of Washington

2020

Committee:

Eric Klavins

Azadeh Yazdan-Shahmorad

Program Authorized to Offer Degree:

Bioengineering

University of Washington

Abstract

Predicting expression levels of *de novo* protein designs in yeast through machine learning.

Patrick Slogic

Chair of the Supervisory Committee:
Eric Klavins
Department of Electrical and Computer Engineering

Protein engineering has unlocked potential for the biomolecular synthesis of amino acid sequences with specified functions. With the many implications in enhancing therapy development, developing accurate functional models of recombinant genetic outcomes has shown great promise in unlocking the potential of novel protein structures. However, the complicated nature of structural determination requires an iterative process of design and testing among an enormous search space for potential designs. Expressed protein sequences through biological platforms are necessary to produce and analyze the functional efficacy of novel designs. In this study, machine learning principles are applied to a yeast surface display protocol with the goal of optimizing the search space of amino acid sequences that will express in yeast transformations. Machine learning networks are compared in predicting expression levels and generating similar sequences to find those likely to be successfully expressed.

ACKNOWLEDGEMENTS

I would like to thank Dr. Eric Klavins for accepting me as a master's student in his laboratory and for providing direction and guidance in exploring the topics of machine learning and recombinant genetics throughout my graduate program in bioengineering. I appreciate his feedback on directions of study and pointed advice in searching for background in methodologies presented in this research. I would also like to thank the Ph.D. and postdoctoral students in the Klavins Lab and in the Institute for Protein Design for providing data, code examples, and insight into the research methodologies of protein engineering. I would like to give a particular mention to Devin Strickland from the Klavins Lab and Hugh Haddox from the Institute for Protein Design for their explanation of available datasets and yeast expression technologies. I would also like to thank the members of the DARPA Synergistic Discovery and Design Environment (SD2E) program for access and collaboration on developing analyses extending from experimental data. Lastly, I would like to thank my professors at the University of Washington who provided guidance and explanation of advanced topics in bioengineering throughout my graduate program.

TABLE OF CONTENTS

LIST OF FIGURES

variables and Y. The red dots indicate the observed values for specific quantities, which further display the difference between the observed values and the estimated values based on the functional relationship between the inputs and response. B) The plot of mean-squared error (red), bias (blue), and variance (orange) of the statistical learning algorithm displays the bias-variance tradeoff in attempting to fit a predictive model to training data while minimizing predictive loss on unseen test data. The dashed line represents the irreducible error of a predictive function. [76]

Figure 11. Representation of the core components of deep learning design. The objective function quantifies the performance of the network on a specified task. Learning involves searching for the synaptic weights, through a set of updating rules, that maximize or minimize the objective function, quantified by a loss or cost measure. The network architecture specifies the connection of these units and determines the flow of information. [74]

Figure 12. Multilayer networks and backpropagation. A) Forward pass transforms the input data through hidden representation into the dimensions of the output units. B) Backward pass updates the weights through the gradient of errors with respect to the weights. [77]

Figure 13. Overview of different inputs and applications in biomedical and health informatics where statistical learning algorithms can ascertain unknown connections between biological signatures and specified therapeutic goals. [78]

Figure 14. Generalized architecture for machine-learning based amino acid encoding methods. The input is the original sequential amino acid encodings, the output is the desired experimental measure of fitness relating to sequence, and the hidden layer represents the new numerical encodings of corresponding amino acids. [85]

Figure 15. Model architecture for dense layer activation in Tensorflow. A) Visualization of input, hidden, and output layers. The input vectors of flattened sequence representations are connected to each neuron of subsequent layers with decreasing dimensional size. The values established in the tensor of each layer are the result of linear connections, activation functions, and regularization. B) Layer specifications in Tensorflow. [99]

Figure 16. Bidirectional LSTM network architecture. A) Visualization of information flow for embedded inputs. B) Layer specifications in Tensorflow. [99]

Figure 17. Convolutional network representation. A) The information flows through the convolutional network, incorporating encoded amino acid sequences and associated measures of performance. B) The skip connection adds the input information for the convolutional block to the output of convolutional operations. [99]

Figure 18. Model architecture of the convolutional neural network in Tensorflow.

Figure 19. Encoded matrices of amino acid sequences. A) One-hot encoding matrix (65x21). B) Hydropathy difference matrix (65x65). C) Language embedding matrix (65x128).

Figure 20. Amino acid code frequencies for training, validation, and total dataset.

Figure 21. Architecture of variational autoencoder. The encoder compresses the input data down to two latent variable representations. The decoder samples from the latent space and transforms them back to the original dimensions of the input data. Together, these two architectures form the overall structure of the variational autoencoder (VAE).

Figure 22. Ranked predicted probability of sequence enrichment among expressing proteins, colored by topology class.

Figure 23. Plot of predicted expression ratio (ratio of expressed counts to naive library counts) through CNN compared to true values. The red line indicates where the two values are equal to one another.

Figure 24. Comparison of predicted stability score to predicted expression ratio, colored by topology class.

Figure 25. Latent space representation of aggregated Rosetta designs colored by topology classification.

Figure 26. Latent space representation of aggregated data in variational autoencoder. A) All Rosetta designed sequences colored in blue. Observed sequence colored in red. Single mutation variants of observed sequence colored in yellow. B) Same plot with all other sequences colored according to k-means cluster labels.

Figure 27. Reconstructed sequences from latent space represented by one-hot encoding of each residue position. The top figure represents the one-hot encoding of each position in the amino acid sequence. The positions indicating the amino acid type are the same at all positions except for the one where the mutation occurred. The position of the original amino acid type is colored in red, while the one indicating the type of the mutated sequence is yellow. The middle two plots indicate the probability weight matrix for both the reference sequence and the mutated sequence. These weight matrices were the result of decoding the latent space representations of each sequence. The bottom figure represents the difference in the values of the probability weight matrix between the reference sequence and the mutated sequence.

Figure 28. Plot of principal component analysis of mutated sequences on latent space representation. A) View of overall latent space plotted with PCA eigenvector (in turquoise) of mutated sequences. B) Zoomed in view of mutated sequences. Each dot along the eigenvector represents a small perturbation in latent space to test for decoded output.

Figure 29. Reconstructed sequences from latent space representation. The top plot displays the original sequence's reconstruction. The middle plot displays the reconstruction following a perturbation in latent space along the eigenvector of PCA analysis. The bottom plot displays the difference in reconstructed predictions between the reconstructed original sequence and the perturbed reconstruction.

Figure 30. The left shows the sequence reconstruction based on the location within the latent space of the trained variational autoencoder. Each perturbation corresponds with a position-weight matrix used to predict the most likely sequence given the coordinates in latent space. The right shows the movement within the latent space representation. The color of the background corresponds to the log probability of decoding the original reference sequence within the latent space representation.

Figure 31. Two-dimensional latent space representation of the final layer of the trained convolutional neural network. The reference sequence is colored in red, the sequences of single mutations are colored in orange, and the remaining sequences are colored according to the values of their predicted expression ratios as output from the CNN.

Figure 32. The impact of single-residue mutations on predicted expression enrichment. The plot shows the probabilities from the trained CNN binary classifier, with the reference sequence colored in red and all sequences with single-residue mutations colored in orange.

Figure 33. A) Locations of top 10 most highly expressed sequences for each of the 22 provided topologies represented in the VAE's latent space shown in red. B) Small perturbations away from the locations of the most highly expressing proteins, 400 for each topology, in latent space shown in blue.

Figure 34. The predicted expression metrics of the top 10 expressed sequences in the class "0a6b." In both plots, the top 10 most highly expressing sequences are colored in red. The other sequence predictions are from decoded perturbations in the variational autoencoder's latent space. A) The predicted binary probability of an increased presentation among expressing proteins for the sequence reconstructions. B) The predicted expression ratio of expressing counts to naive counts for the sequence reconstructions.

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Proteins form the foundation of biological processes that make all life possible. The fates of biological cells are controlled by interacting proteins in metabolic and signaling pathways, in complexes such as the molecular machines that synthesize and use energy sources such as adenosine triphosphate (ATP), in the replication and translation of genes by polymerases, or building up the cytoskeletal infrastructure [1]. Understanding the structural outcomes of proteins translated from the genome has captured the geometrical determinants of a protein's capabilities in a biological environment. Function prediction from structure has been achieved both through global comparison of naturally occurring protein homologies and the targeted binding analysis with specified complementary structures [2]. With the completely sequenced genomes of more than 600 organisms, the computational analysis of the molecular function, biological process, and cellular components has created data for analyzing protein data down to the atomic level.

By decoding the information encoded in the complex structural and biochemical aspects of natural proteins, the generation of *de novo* protein structures mimicking and enhancing their naturally observed properties has become a tangible possibility. In targeted therapeutics, *de novo* design offers the promise for creating binding proteins with shapes customized to bind to targeted disease pathways [3]. Sequence similarity of homologous sequences in which at least one's structure and function has been experimentally determined has allowed researchers to characterize families of proteins in which an unknown sequence is likely to show similar structural and functional outcomes. Evolutionary development has explored the viability of natural proteins that now offer a powerful toolkit as references for engineering new attributes and functions in new protein architectures [4].

However, expanding the space of proteins beyond those isolated from natural organisms has proved challenging due to the complexity of interactions among amino acid units and with the environment in which a translated protein folds. The promise of theoretical designs suffer from a low percentage of designs having NMR spectra and temperature melts of tightly packed proteins [5]. Cooperative impacts of amino acid pairs in a sequence can alter the interactions of the entire sequence and undermine the actual structural outcome of designs based on proteins found in nature. While computational methods have captured many trends in the protein folding landscape, there has yet to be developed a process for designing sequences with reliable structural outcomes without extensive experimental validation.

In generating amino acid sequences that adopt a stable three-dimensional structure through adaptations of existing scaffolds or through statistical modeling of new structural motifs, high throughput methods are required for analyzing the cascade of effects resulting from alterations to naturally validated proteins. Recombinant genes provide a platform for inserting DNA fragments with codon specificity for a particular amino acid sequence into a host genome for the expression of these fragments through natural cellular pathways. While the molecular interactions of amino acid sequences ultimately contribute to their subsequent structure and functions with other biomolecules, the expression of these proteins is crucial to creating high-throughput analyses of design alterations in an enormous search space for potentially successful designs. Experimental validation of these designs requires time-consuming and costly resources which compound as the number and variety of tested designs are increased. Analyzing the expression of *de novo* designs offers both a parallel insight into the successful fitness of a

design as well as the means to screen potential designs in the iterative process of successfully creating proteins with desired properties.

1.1 Protein Composition

The primary structures of proteins are composed of sequences of variable lengths, where each component in the sequence is one of 20 amino acids. While there exist in nature over 500 amino acids, only 20 of them are proteinogenic α-amino acids that are encoded by the universal genetic code [6]. These amino acids contain two functional groups, an amino group ($-NH_2$) and a carboxyl group (-COOH), attached to the central α-carbon. Additionally, this central carbon is bonded to a hydrogen atom and side chain, known as an R-group, providing the chiral nature of the amino acid's α-carbon. Multiple amino acids come together through peptide bonds in which dehydration reactions take place to bond the COO^- group of one amino acid and the NH_3^+ group of another, forming sequences of amino acids proceeding from a polypeptide's N-terminus to its C-terminus (Fig. 1).

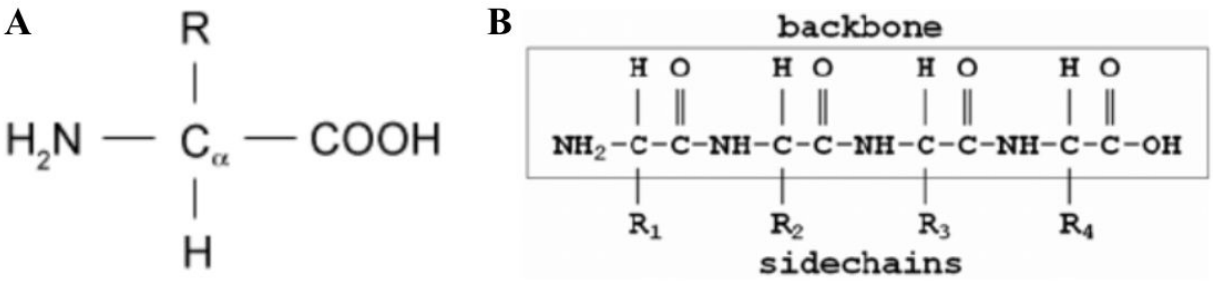


Figure 1. A) Fischer projection for general representation of amino acid molecules. The central α-carbon is bonded to an amino group, a carboxyl group, a hydrogen atom, and the R group unique to each amino acid. B) Dehydration reactions between the carboxyl group of one amino acid and the amino group of another form peptide bonds that link multiple amino acids together, establishing a protein's carbon backbone. [7]

The functional side chains, the compounds attached as the R-group to the α-carbon of each amino acid, determine much of the unique structure and subsequent biochemical behavior of polypeptide chains. The 3-dimensional conformation of this central carbon's atomic bonds to four different constituents produce the optically active stereoisomers, the L-configuration and D-configuration, of each amino acid in which one enantiomer is the mirror image of the other (Fig. 2). Furthermore, the tendency for the amino group and carboxyl group, on opposite ends of each chiral carbon, to gain or lose a proton produce molecules, known as zwitterions, that can behave as both an acid and base depending on the protein's environment, regardless of the attached functional groups [15]. This repeated stereocenter provides the reference for analyzing the bond angles and electron density distribution throughout an amino acid sequence [14]. With a chiral, stereogenic central carbon in each amino acid, the functional R-group of each amino acid provides its distinct interactions with other functional side chains and extraneous molecules for a protein's carbon backbone [10].

Figure 2. Representation of chiral α-carbon and the stereoisomers, the L-configuration and D-configuration, as a result of constituent connectivity. The opposite charges of the amino and carboxyl groups under physiological pH display the zwitterion arrangement providing both acidic and basic properties to each amino acid. [11]

Each amino acid can be classified based on biochemical properties of its functional group (Table 1). The structures of the observed functional groups are broadly classified into five

categories: nonpolar/nonaromatic, aromatic/uncharged, polar/nonaromatic, negatively-charged/acidic, and positively-charged/basic. Among the various biochemical characteristics of each amino acid, their isoelectric points and hydropathy are important indicators of their unique interactions in solvent (Table 1). The isoelectric point refers to the environmental acid/base conditions resulting in the deprotonation of the amino and carboxyl groups, where the amino acid exists in its zwitterionic form [11]. For acidic amino acids, the isoelectric point lies between the pH levels in which the carboxyl group and the side chain become deprotonated, whereas the isoelectric point for basic amino acids lies between the pH levels in which the side chain and amino group become deprotonated. The hydropathy index is a quantitative measure of the degree of hydrophobicity characterized by numbers representing hydrophobic moments. A larger (more positive) hydropathy index indicates more hydrophobicity, the tendency for that amino acid to avoid an aqueous environment due to its polarity. Molecules with a similar hydropathy index have an affinity for one another, whereas those with largely different hydropathy indexes will repel one another [12].

Amino Acid	3-Letter Code	1-Letter Code	Side Chain Properties	Side chain groups	Isoelectric Point	Hydropathy Index
Alanine	Ala	A	Nonpolar / Nonaromatic	Alkyl	6.11	1.8
Arginine	Arg	R	Positively-charged / Basic	Guanidino	10.76	-4.5
Asparagine	Asn	N	Polar / Nonaromatic	Carboxamide	5.43	-3.5
Aspartic Acid	Asp	D	Negatively-charged / Acidic	Carboxylate	2.98	-3.5
Cysteine	Cys	C	Polar / Nonaromatic	Thiol	5.15	2.5
Glutamic Acid	Glu	E	Negatively-charged / Acidic	Carboxylate	3.08	-3.5
Glutamine	Gln	Q	Polar / Nonaromatic	Carboxamide	5.65	-3.5
Glycine	Gly	G	Nonpolar / Nonaromatic	Hydrogen	6.06	-0.4
Histidine	His	H	Positively-charged / Basic	Imidazole	7.64	-3.2
Isoleucine	Ile	I	Nonpolar / Nonaromatic	Alkyl	6.04	4.5
Leucine	Leu	L	Nonpolar / Nonaromatic	Alkyl	6.04	3.8
Lysine	Lys	K	Positively-charged / Basic	Primary amino group	9.47	-3.9
Methionine	Met	M	Nonpolar / Nonaromatic	Alkyl / Sulfur	5.71	1.9
Phenylalanine	Phe	F	Aromatic / Uncharged	Benzyl	5.76	2.8
Proline	Pro	P	Nonpolar / Nonaromatic	Cyclic secondary amine	6.30	-1.6
Serine	Ser	S	Polar / Nonaromatic	Alcohol	5.70	-0.8
Threonine	Thr	T	Polar / Nonaromatic	Alcohol	5.60	-0.7
Tryptophan	Trp	W	Aromatic / Uncharged	Heterocyclic indole	5.88	-0.9
Tyrosine	Tyr	Y	Aromatic / Uncharged	Phenol	5.63	-1.3
Valine	Val	V	Nonpolar / Nonaromatic	Alkyl	6.02	4.2

Table 1. Amino acids with 3-letter codes, 1-letter codes, side chain properties, isoelectric points, and hydropathy values. [6] [8] [13]

1.2 Structural Determinants

Due to the complexity of interactions within a protein's framework, a significant variety of overall conformations are made possible from the combination of these 20 amino acids (Fig.

3). The linear arrangement of amino acids determines each protein's primary structure. Coding of the protein's sequence from the N-terminus, the amino end, to the C-terminus, the carboxyl end, is encoded through the sequence of three-nucleotide codons in an organism's DNA. This primary structure is stabilized by the covalent peptide bonds between adjacent amino acids. The proximity of amino acids to one another in a linear sequence is an initial factor in the segmentation of hydrophilic/hydrophobic regions across the protein and the formation of local constraints for side-chain orientations [13].

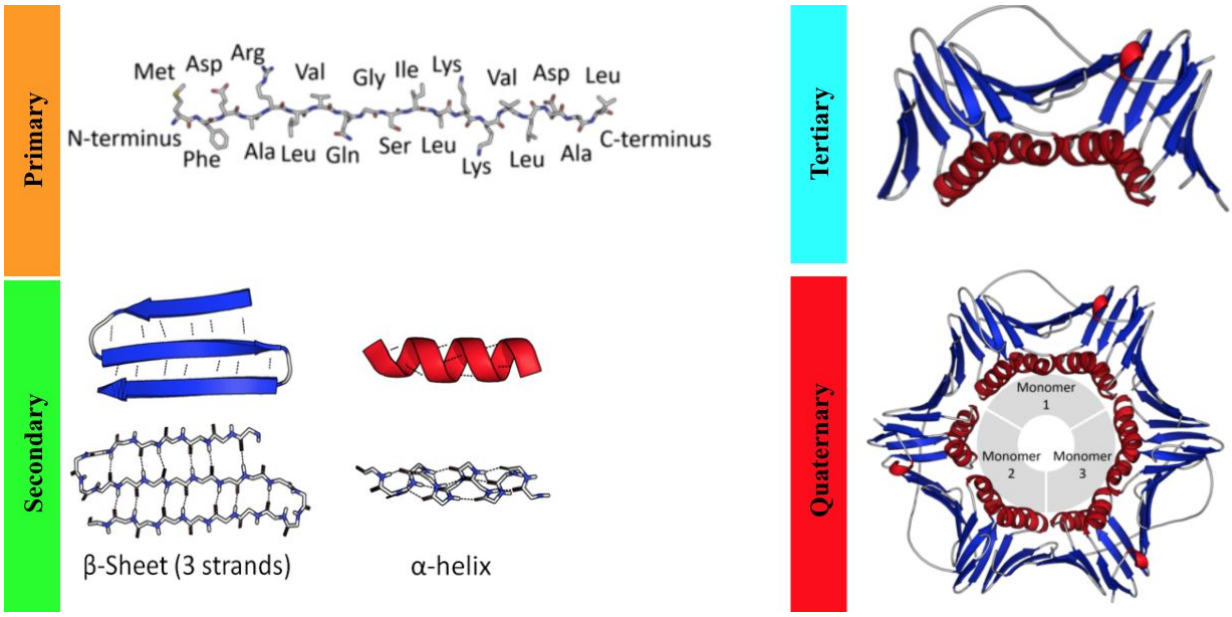


Figure 3. Representation of structural hierarchy of human PCNA DNA-binding protein. The primary structure is defined by the sequence of amino acids from the N-terminus to the C-terminus. The secondary structure is determined by the local structure of neighboring amino acids, forming common fragment motifs seen in natural protein structures. The tertiary structure is defined by the three-dimensional shape into which the sequence folds. The quaternary structure is defined by the functional form of multiple interacting polypeptide monomers. [18]

A protein's secondary structure, the local three-dimensional of neighboring amino acids, is largely the result of hydrogen bonding between nearby units. The most common secondary structures are α-helices and β-pleated sheets. The α-helix is characterized as a rod-like structure

with a clockwise peptide coiling around a central axis, stabilized by hydrogen bonds between a carbonyl oxygen atom and an amide hydrogen atom four residues down the chain [21]. β-pleated sheets, arranged in either a parallel or antiparallel arrangement, contain peptide chains alongside one another in strands held together by intramolecular hydrogen bonds between a carbonyl oxygen on one chain and an amide hydrogen on an adjacent chain [22]. More flexible loop structures connect these defined structures along a protein's sequence. Several less common structures are present in native proteins, and other extended structures, such as the polyproline helix and α-sheet, are hypothesized as important folding intermediates [20]. When characterizing secondary structure elements, the DSSP system (hydrogen bond estimation algorithm) assigns one of eight letters, within three broad categories, to each amino acid along a sequence based on an energy function relating its distance from other carbonyl oxygens and amide hydrogens [21]. Helical fragments are described with three letters: G (3_{10} helix), H (α-helix), and I (π-helix). β-strands are described as either E (β-bridge) and B (β-bulge). Loop fragments are described with three letters: S (bend), T (hydrogen-bonded β-turn), and C (unclassified with no regular hydrogen bonding) [25].

A protein's tertiary structure, its three-dimensional shape mostly determined by hydrophilic/hydrophobic and acid-base interactions between side chain groups, impacts the short and long-range electrostatic forces determining each residue's secondary structure [23]. The secondary structure seen in the native conformation of a protein reflects an energetic compromise between the local conformational propensities and global restraints emerging from close packing, specific patterns of side chain interactions, and hydrogen bond restraints [23]. Short-range interactions are determined by both general and sequence-specific local statistical

potentials from the lattice structure of atoms surrounding a residue's α-carbon. Long-range interactions are largely determined by centrosymmetric properties and hydrophobic burial potentials [23]. Upon folding into their native state three-dimensional structures, multiple polypeptide chains can further conform to quaternary structures. Protein-protein interactions of these tight complexes, such as the binding of ribonuclease inhibitor to ribonuclease A to protect RNA degradation, can be engineered to favor certain oligomerization states that control biological pathway mechanics [24].

Efforts in protein engineering have led to statistical representations of these interactions in modeling dynamics. Characterizing the dihedral angles between amino acids along a sequence backbone has provided a cartesian representation of atoms in a protein's projected folded conformation (Fig. 4). Based on the orientation of repeating α-carbon (C_α), secondary carbon(C'), and nitrogen atoms in the backbone, the rotational angles around the N-C_α bond (phi angle) and the around C_α-C' bond (psi angle) allow for restricted orientations in which potential amino acids must conform to produce a stable design [28]. The structural determination of *de novo* sequences based on Euclidean constraints, such as the characterization of backbone motions through four parameters in Crick's Parameterization, have shown efficacy in reproducing many *in silico* designs [29]. Electrostatic calculations extending from these geometric constraints have allowed for the probabilistic representation of changes in free energy at positions in a sequence based on enthalpy and entropy changes of residue substitutions, such as the free energy determination of helical stability based on the Zimm-Bragg model for statistical weighting of combinatorial residues [21][30].

Figure 4. Ramachandran plot of phi and psi angles, displaying characteristic regions of structural elements based on geometric constraints of sequence backbone. [31]

1.3 Biological Investigations in Proteostatic Mechanisms

Tracing the pathway from genetic code to functional protein structure was inspired by Dr. Christian Anfinsen's scientific credo that the sequence of amino acids and the environmental conditions in which folding occurs are sufficient to determine a protein's 3-dimensional structure [32]. Dr. Anfinsen's initial studies into determinants of the protein ribonuclease A's structure established a framework for protein folding experiments. The undertaking in establishing determinants of protein structure was rooted in the observation that, despite the fact that a particular amino acid sequence could potentially adopt a vast number of conformations, a protein defies entropy and collapses into a specific ordered structure [32]. This energetic benefit from sequestering hydrophobic amino acid side chains from the polar environment of a biological cell

is achieved through a cascade of intermediates between the DNA coding of proteins to their 3-dimensional functional forms.

Expressed proteins must navigate a cascade of dense protein regions and varying degrees of polarity as they emerge from the ribosomes before becoming a functional member of the cellular environment. The exposure of a translated sequence to the cytosol, organelles, and other proteins within the cell can cause unfolded portions to become tangled with their neighbors and expose the hydrophobic regions, risking the formation of tangled aggregates [40]. The fact that cell membrane proteins, containing large stretches of hydrophobic residues presented on the cell surface to remain anchored in the fatty acid tails of membranes, do not unfold and aggregate after passing through a polar environment hinted at the presence of cellular mechanisms mediating the folding process. Investigations into further folding processes across cellular membranes in yeast led Dr. Franz-Ulrich Hartl and Dr. Arthur Horwich to discover the first folding micromanager in the chaperone protein heat shock protein 60 (HSP60). In studying the deficiency of the enzyme ornithine carbamoyltransferase implicated in diseases related to the urea cycle, HSP60 was discovered as a crucial component of ATP-dependent protein translocation and refolding as they cross the mitochondrial membrane (Fig. 5) [39]. Following this chaperone discovery, the folding cascade process was confirmed in the discovery of the catalytic bacterial equivalent of HSP60 known as GroEL [39]. This family of molecular chaperones acts to encapsulate unfolded proteins through multiple protein complexes, creating an environment of selective hydrophobic interactions to prevent aggregation, for folding following mitochondrial import (Fig. 5).

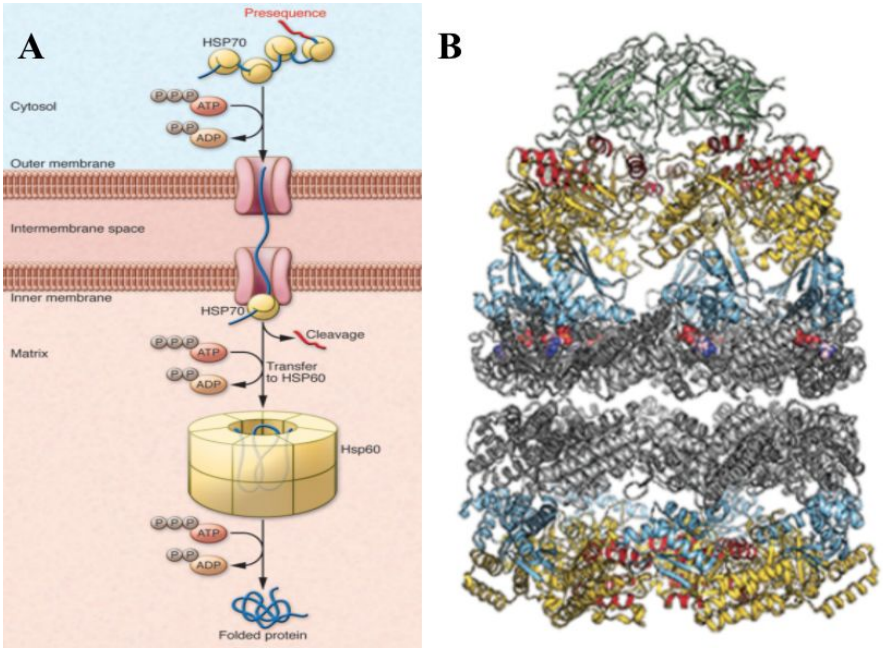


Figure 5. A) Protein transport into the mitochondria through the ATP-driven chaperone activity of HSP60 and HSP70. B) GroEL chaperone protein encapsulates an unfolded protein around itself to prevent hydrophobic sequences from aggregating in crowded cells. A capping protein isolates the unfolded chains and makes the chaperone core polar so the protein can fold. [41]

Since their original discovery, a multitude of both ubiquitous and specified chaperone proteins have been shown to contribute structural and kinetic advantages in the dynamic process of DNA transcription to protein expression. From the pervasive heat shock protein 70, greeting nascent chains that come off the ribosomes to regulate solubility through the ubiquitin proteasome system, to the bacterial Trigger Factor binding to several regions of alkaline phosphatase, variable regions of amino acid sequences achieve their funcional state through a continuous cascade of intermediate bonding configurations [33][34]. Their roles in the expression pathway further allow for the integration of mutations in the evolutionary pathway of cellular functionality. Chaperones enable evolution and de novo design expression by stabilizing mutations while the cell explores the benefit of the mutations [36].

The dynamic involvement of intermediate proteins led to understanding of proteostasis, in which competing pathways control the folding, trafficking, and degradation of proteins in the cell (Fig. 6). By controlling the conformation, binding interactions of quaternary structure, and location of individual proteins, proteostasis allows for differentiated cells to change their physiology for development [37]. The 3-dimensional conformation and binding interactions of a particular sequence can vary significantly depending on its state in the proteostatic pathway. Temporal adaptation of these proteostatic mechanisms is necessary to manage a constantly changing proteome as new proteins enter the translation pathway and misfolded proteins gather in the cell [37].

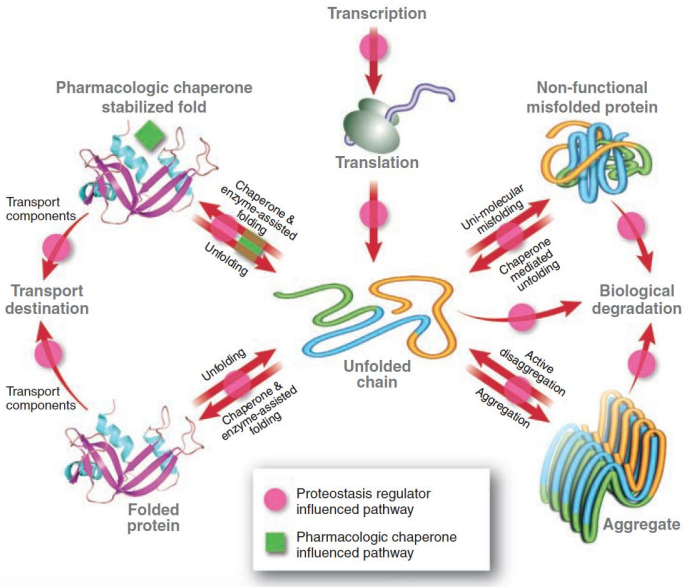


Figure 6. Proteostasis network maintaining the proper folding, trafficking, and degradation of translated proteins while monitoring their impact on the overall cellular network. [37]

These interacting pathways present both causes of diseases and targets for disease intervention. For instance, Alzheimer's Disease has presented the toxic byproduct of a dysfunctional proteostatic pathway in the presentation of misfolded protein aggregates as

amyloid fibrils [42]. While the multifaceted steps in folding pathways make it difficult to specifically target the misstep leading to negative folding outcomes, potential therapeutic targets lie in the target bonding or replacement of proteostatic mechanisms. Targeted binding interactions with the Hsp90-Cdc37-Cdk4 kinase complex and the adenosine triphosphate pocket providing enzymatic energy have been proposed in targeted therapeutics, such as the role of Tafamidis in targeting amyloid neuropathy [35].

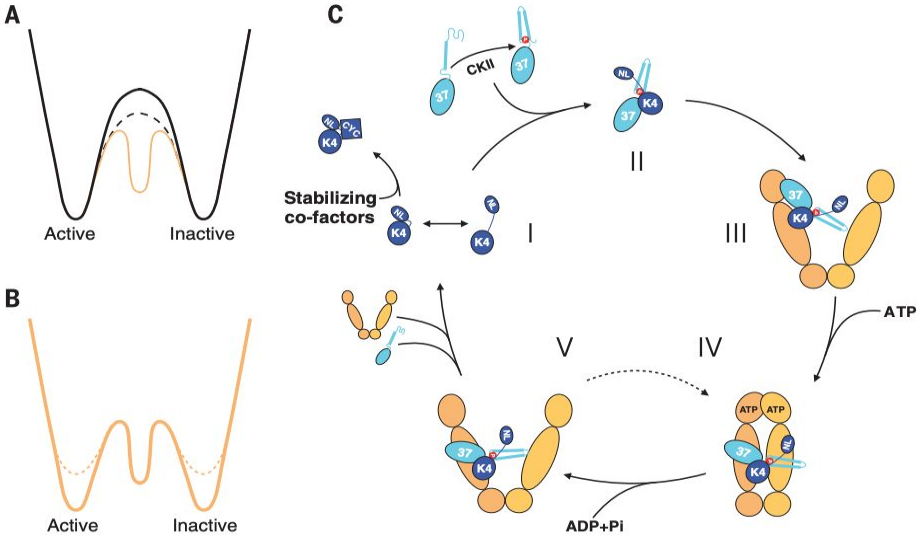


Figure 7. Model for linkage between kinase folding and activation for proposed model for Hsp90-Cdc37-Cdk4 kinase complex cycle. A) Transition between states through an unfolded intermediate (dashed line) presents a lower energy barrier. B) Comparison of the active and inactive states between nonclients and client kinases. C) Model for regulation of kinase cycle. [35]

1.4 Yeast Expression

Targeting proteostatic pathways for cellular intervention or the synthesis of novel molecular structures has motivated the design of *de novo* protein sequences. While the study of

protein folding *in vitro* has revealed the dynamic interactions of amino acids with each other, the actualized formation of these proteins depends on the complex biological environment found in nature. The parameterization of forces shaping protein expression and evolution necessitates a biological environment in which reproducible mechanisms of the proteostatic system influence a protein's eventual conformation. Biological expression systems offer the ability to analyze the translated products of recombinant DNA technology. High throughput systems allow for the comparison of large amounts of protein sequences in targeted design of protein characteristics, yet their applicability is limited by their ability to express large amounts of active protein [46]. The anatomical and biochemical characteristics of different cells affect the efficiency of transcription and translation.

Several expression systems have been developed with the aim of producing functional amino acid sequences with reasonable cost and effort. Bacterial, yeast, insect, and mammalian cells have been commonly utilized for expressing large amounts of active protein (Table 2). No system represents a universal expression toolset that can guarantee high yields of recombinant product as every polypeptide product poses its own biomechanical and energetic obstacles in terms of expression [46]. Bacterial cells, particularly *Escheria Coli*, were initially utilized for their ability to produce protein in large quantities, providing the opportunity to purify, analyze, and use expressed protein in a short amount of time [46]. These prokaryotic cells play an important role in producing fragments that do not require glycosylation through extracellular expression of protein sequences through the periplasm of their cell membranes, yet their yields are highly dependent on the sequence in question [46]. Eukaryotic cells offer an advantage in their biological relevance to therapeutic design, providing advanced protein folding pathways for

secreting correctly folded heterologous proteins into culture media [46]. Insect cells provide a popular pathway for baculovirus-mediated gene expression while maintaining the functional activity of a foreign protein of interest, yet their considerably higher oxygen demand and insufficient recognition of certain proteolytic cleavage sites can create stress on the translation pathways in sensitive cells [46]. Mammalian cells offer the natural mechanisms for recognizing and processing eukaryotic proteins, yet the significant variance of their transcriptional elements in different cell lines and complex growth medium requirements limit the throughput of tested designs [46].

Characteristics	*E. coli*	Yeast	Insect cells	Mammalian cells
Cell growth	rapid (30 min)	rapid (90 min)	slow (18-24 h)	slow (24 h)
Complexity of growth medium	minimum	minimum	complex	complex
Cost of growth medium	low	low	high	high
Expression level	high	low-high	low-high	low-moderate
Extracellular expression	secretion to periplasm	secretion to medium	secretion to medium	secretion to medium
Posttranslational modifications				
Protein folding	refolding usually required	refolding may be required	proper folding	proper folding
N-linked glycosylation	none	high mannose	simple, no sialic acid	complex
O-linked glycosylation	no	yes	yes	yes
Phosphorylation	no	yes	yes	yes
Acetylation	no	yes	yes	yes
Acylation	no	yes	yes	yes
gamma-Carboxylation	no	no	no	yes

Table 2. Comparison of expression system components for inserted nucleic acid vectors and posttranslational modifications. [44]

Yeast cells present highly efficient secretion of folded and processed heterologous proteins on simple growth media as compared to other eukaryotic expression platforms (Fig. 8). As the most well-known yeast strain and first eukaryotic genome to be completely sequenced, *Saccharomyces cerevisiae* represents a well-characterized, efficient, and robust cellular platform for a variety of expression purposes [47]. The two primary vectors used for the expression of

genes in yeast are episomal vectors that propagate extrachromosomally and integrated vectors where chromosomal integration occurs by homologous integration [46]. Classical mutagenesis can be used for increasing homologous expression, while recombinant production can be facilitated by introducing a gene of interest in autonomously replicating plasmids or integrating it in the genome. The designs of a yeast expression system for a desired protein product resulting from an inserted vector require an origin of replication or integration, a strong promoter, and a selection marker [47].

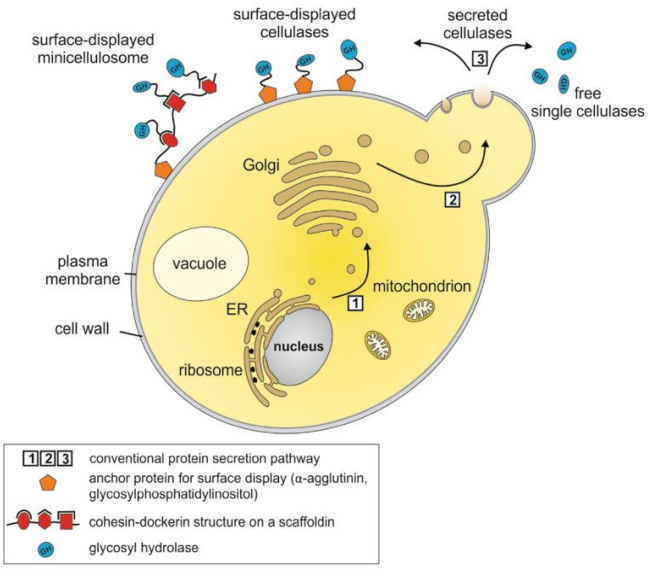


Figure 8. Single yeast cell showing the main cell compartments involved in recombinant protein expression. [50]

Vectors applied for transformation of yeast host strains are hybrids of bacterial and yeast-derived sequences [47]. The bacterial fragment carries elements crucial for plasmid proliferation in *E. coli*, such as the selection marker indicating the success of transfection and the origin of replication. Expression of recombinant proteins in *Saccharomyces cerevisiae* can be done using three types of vectors: integration vectors (YIp), episomal plasmids (YEp), and

centromeric plasmids (YCp) [51]. YIp integrative vectors do not replicate autonomously, but they are integrated into the genome loci at low frequencies through homologous recombination [51]. These vectors typically integrate as a single copy, yet methods to increase copies of gene integrations such as targeting the rDNA cluster have been developed for over-expressing certain genes. YEp episomal vectors replicate autonomously due to the presence elements of the 2 μm yeast plasmid. These vectors can exist in cells in over 30 copies, yet they can lead to unstable strains with significant batch variation in the production process [47]. YCp centromere vectors, typically used in low-level expression and as cloning vectors, are autonomously replicating single copy plasmid vectors containing centromere sequences and autonomously replicating sequences [51].

Selection markers are classified into complementation markers and dominant selection markers. Complementation markers are marker genes that complement an auxotrophic mutation in the genome, such as URA3 or LEU2, used in selection for all expression systems. Dominant selection markers are antibiotic markers such as G418 or cyclohexamide [47]. Selection of transformed *E. coli*, typically using an antibiotic-resistant marker, allows for the isolation of DNA or a plasmid before subsequent transformation and selection of *S. cerevisiae* to ensure plasmid persistence through generations. Promoters, which are DNA sequences that can recruit transcriptional machinery and lead to transcription of the downstream DNA sequence, are typically utilized for the overexpression of heterologous proteins [47]. Constitutive promoters, such as the ADH2 promoter, remain active under all circumstances in the cell, whereas inducible promoters, such as the inducible HXT7 high affinity hexose transporter, are active with the presence of an inducible stimulus [52]. The isolated DNA sequence is transformed in the yeast

cells through shuttle vectors designed with the promoter, a cloning site for gene insertion, and regulatory elements for maintenance [51]. Along with providing a carbon-source is necessary for the growth and induction of product for each yeast cell phenotype, the glycosylation of sequences to form mannoproteins proves a crucial step in proliferating expression across different phenotypes [53]. Glycosylation and several other post-translational modifications, such as phosphorylation and N-acetylation, are enacted on a translated protein during its expression in the yeast cell surface to modulate protein activity and reduce degradation [54].

Following the selection and scaling-up of high expressing yeast clones in appropriate culture media, fluorescent proteins and epitope tags commonly integrated into the chromosomal locus offer a reporting mechanism for tracking a protein's cellular dynamics [56]. Flow cytometry can be used in cell sorting to pass cells with fluorescent markers through a light beam to measure the light scatter characteristic of each cell's components, allowing for the separation of cells that meet a fluorescence threshold from non-displaying cells [58]. These cells with tagged protein sequences are isolated and purified from the cellular mixture for analysis. Accounting for protease release during cell lysis through protease inhibitors, precipitated proteins can be separated according to amino acid properties such as isoelectric point through ion exchange chromatography, molecular weight through sodium dodecyl sulfate-polyacrylamide gel electrophoresis (SDS-PAGE), or molecular conformation through affinity chromatography. Processing of deep sequencing data identifies specific amino acid sequences through recognition of a unique barcode and cut sites upstream and downstream of an ordered sequence [60]. Genome-wide datasets based on the expression profiles of recombinant vectors have created gene interaction networks to identify central genes in a disease-specific network (Fig. 9) [59].

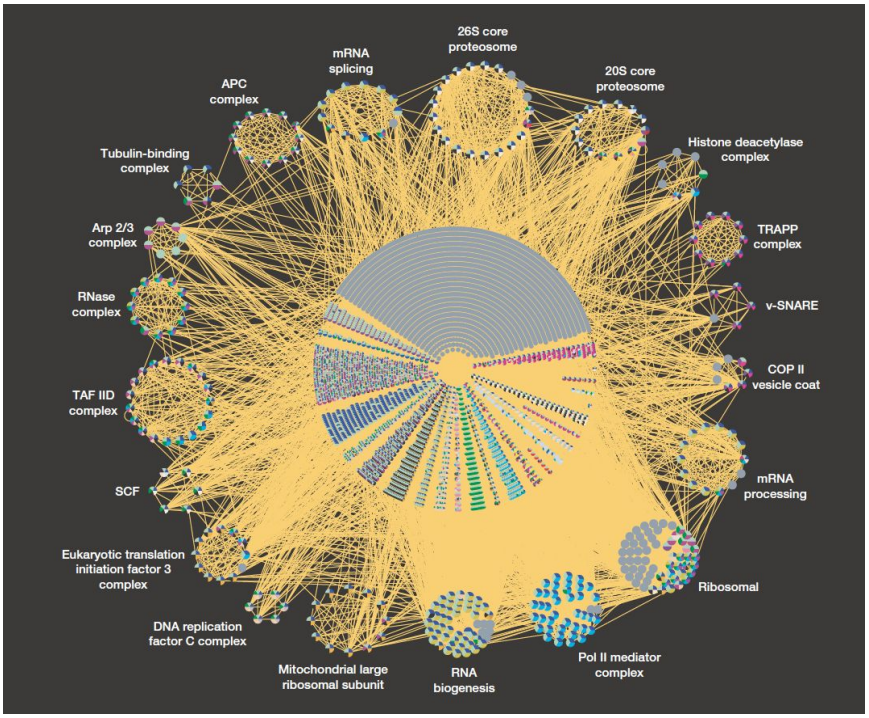


Figure 9. Visualization of combined large-scale interaction datasets extending from targeted expression analysis in yeast. [17]

1.5 Machine Learning Application

Due to the complexity of accurately representing the multitude of interactions that collectively overcome the entropic cost of folding a protein's conformation in a dynamic biological environment, powerful statistical tools are necessary to capture patterns in each protein's extensive structural data. A significant challenge in protein design lies in formulating predictive insights from new, unseen sequential input in order to assess the application of natural constraints on the output of *de novo* designs. The development of advanced computing power

and machine learning methodologies have offered the capacity to generate informed predictions through optimization techniques for complex combinatorial problems [65].

Machine learning broadly encompasses statistical methodologies for programming computers to optimize a performance criterion by using example data or past experience [65]. Its application has shown remarkable ability to make informed, automated predictions in various areas, particularly in the realms of image classification and natural language processing [71]. While the use cases for machine learning models can differ substantially, the statistical relationships inferred through their architectures can be applied effectively to transformed data. Proteins found in nature present a diverse variety of sequential and structural data, where their biochemical properties can be ascertained using modern proteomic informatics tools [17]. Through representing the features of amino acid sequences numerically, each sequence provides an instance which can be mapped to an output measure based on its features and associated feature values [17].

Machine learning methods can accomplish different tasks based on the type of inputs and outputs provided for a model. Supervised learning methods utilize labeled data inputs to finding the functional relationship f between input data x with associated output data, mathematically represented as $y = f(x) + \xi$, incorporating noise terms to account for uncertainty and dimensional transformations for establishing mathematical relationships [71]. In contrast, unsupervised learning methods utilize unlabeled input data with the aim of discovering inherent structures in the data in order to find simplified representations of the input features [71]. Variations of these frameworks have been implemented in machine learning representations that reflect biologically-relevant complexity. In reinforcement learning, which is based on

neuroscientific perspectives of animal behavior, past inputs are generalized to iterative high-dimensional inputs using goal-oriented algorithms at different time steps to maximize cumulative feature reward [73]. In capturing the integration and activation properties of real neurons in the human nervous system, the advent of artificial neural networks has applied interconnected statistical architectures to biological system inputs to model and optimize complex artificial learning systems [74].

The mathematically-defined goal of these statistical learning algorithms is to optimize, through some measure of accuracy, the systematic representation of the information that input variables contain about the associated response variables for use in formulating either predictions or inferences [76]. The qualitative or quantitative variables associated with the expression of designed protein sequences present targeted regression and classification goals for statistical frameworks. Filtering the search space for functions representing the relationship among variables requires a choice of parameters that may be adjusted in order to achieve a model that best fits the training data (Fig. 10). For quantitative outputs, the estimated loss in predictive accuracy is generally assessed with test data's mean squared error, which is a function of model variance, square bias, and irreducible error according to the following [76]:

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

For qualitative outputs, the estimated loss in predictive accuracy is measured by the test error rate between predicted and actual classes according to the following [76]:

$$\mathrm{Ave}\left(I(y_0 \neq \hat{y}_0)\right).$$

The degree of flexibility allowed in the model impacts its predictive accuracy on new, unseen inputs through the amount which the estimated relational function will change if fit to a different training data set. In parallel, the model's bias relates to the error introduced by approximating a complex problem by a simpler model. The two factors tend to be inversely related through a bias-variance trade-off (Fig. 10). Achieving minimal values of both variance and bias is a primary challenge in statistical learning for complex biological processes in that a predictive model needs to simultaneously capture relationships in a dataset while avoiding overfitting such that the model accurately extends to new observations [75].

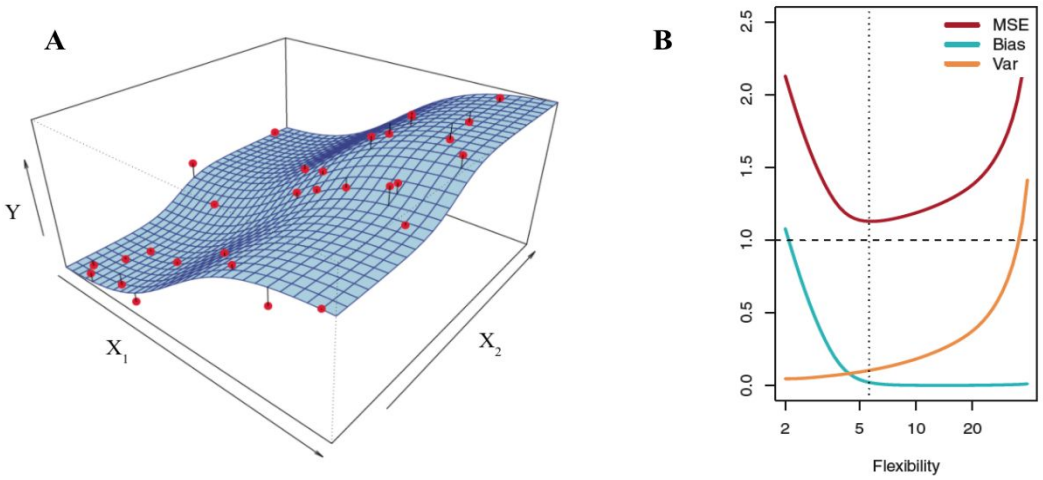


Figure 10. A) The plot displays the desired response variable Y as a function of the input features X_1 and X_2. The blue surface represents the true underlying relationship between the two input variables and Y. The red dots indicate the observed values for specific quantities, which further display the difference between the observed values and the estimated values based on the functional relationship between the inputs and response. B) The plot of mean-squared error (red), bias (blue), and variance (orange) of the statistical learning algorithm displays the bias-variance tradeoff in attempting to fit a predictive model to training data while minimizing predictive loss on unseen test data. The dashed line represents the irreducible error of a predictive function. [76]

Advanced machine learning architectures become necessary for wide datasets in protein engineering applications, in which the number of relevant input feature influences far exceeds the output dimensions [75]. In controlled experimental designs, machine learning makes

minimal assumptions about the data-generating systems, thus removing the need for potentially inaccurate assumptions while capturing relevant features in complicated nonlinear interactions [75]. Representation learning methods allow software systems to be fed raw proteomic data and automatically discover representations needed for classification or detection. Deep learning methodologies encompass multiple layers of representation with non-linear transformations between layers to extract more abstract representations of the input [77]. The core components of a neural network design for extracting representations of protein sequences include an objective function to be maximized or minimized, a set of learning rules expressed as learning rule updates, and the network architecture expressed as pathways and connections for information flow (Fig. 11) [74].

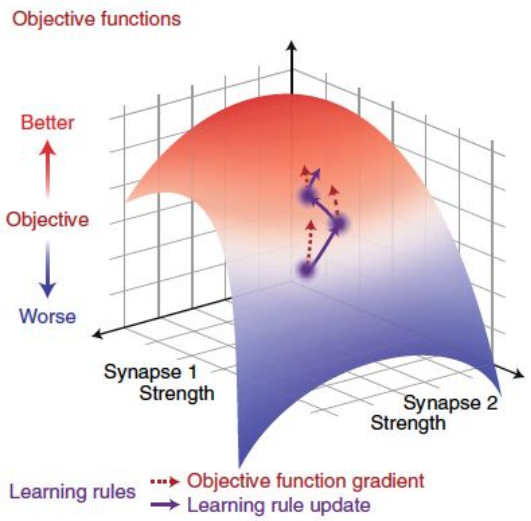


Figure 11. Representation of the core components of deep learning design. The objective function quantifies the performance of the network on a specified task. Learning involves searching for the synaptic weights, through a set of updating rules, that maximize or minimize the objective function, quantified by a loss or cost measure. The network architecture specifies the connection of these units and determines the flow of information. [74]

The natural definition of learning, to change a system that improves its performance, is reflected in the credit assignment procedure for ensuring that changes in parameter weights

produces an improvement in the model's objective function. A learning model's objective function, *F(W)*, offers a measure of the system's performance given a vector of its current synaptic weights, *W*. The weights and subsequent change in performance can be mathematically represented as follows [74]:

$$W \rightarrow W + \Delta W,$$
$$\Delta F = F(W + \Delta W) - F(W).$$

The performance change based on small changes to *W*, assuming that *F* is locally smooth in its current state, can be approximated by the inner product between the weight change and the gradient of F with respect to W as follows [74]:

$$\Delta F \approx \Delta W \cdot \nabla_W F(W).$$

To guarantee an improvement in performance during training, the change in performance is constrained to be a positive value for each step. Gradient-based algorithms operate under the intuition that small steps of weight parameter values in the direction that provides the greatest improvement for that step size. A small step size, η, is chosen to improve the objective function as much as possible as follows [74]:

$$\Delta F \approx \eta \nabla_W F(W)^T \cdot \nabla_W F(W) \geq 0.$$

This challenge of credit assignment in the search for an improved objective function is aided in the architecture of a learning network (Fig 12). Several learning rules have been proposed in providing an estimate of the gradient of the objective function. Simple error back-propagation has proven an effective methodology in training multi-layer architectures through the calculations of stochastic gradient descent [77]. Alternative learning rules, such as feedback alignment, node perturbation, and regression discontinuity design, introduce

significantly higher bias or variance in the learning calculations when compared to error backpropagation [74]. While certain inductive biases can be useful in applying prior knowledge to optimization procedures, features with complex and unknown interactions in protein interactions facilitate a methodology that can ascertain parameter updates through more simple learning updates. This error backpropagation procedure, applied repeatedly through all modules, calculates the derivative of the objective function with respect to the input of a module by working backwards from the gradient with respect to the output of that module [74].

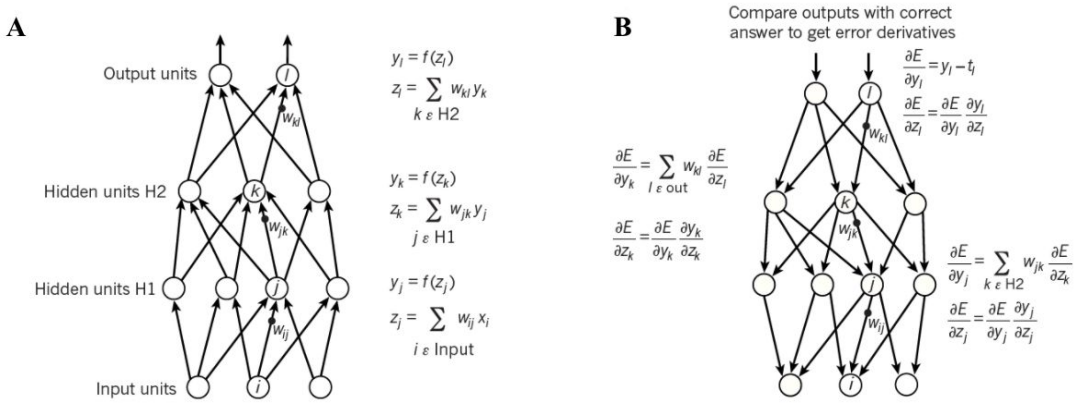


Figure 12. Multilayer networks and backpropagation. A) Forward pass transforms the input data through hidden representation into the dimensions of the output units. B) Backward pass updates the weights through the gradient of errors with respect to the weights. [77]

1.6 Computational Bioinformatics

As software models have increased in complexity and scope of objectives, their applications have extended from simple image classification to large biologically-relevant

datasets (Fig. 13). General areas of machine learning applications in bioinformatics extending from the expression pathway include genomics, pharmacogenomics, and epigenomics [78]. Pinpointing the genetic origin of cellular functions, evaluating variations in an individual's drug response to treatment brought on by a genetic profile, and investigating protein interactions to understand higher level processes have presented challenges in therapeutic development aided by the processing power of high-order algorithms.

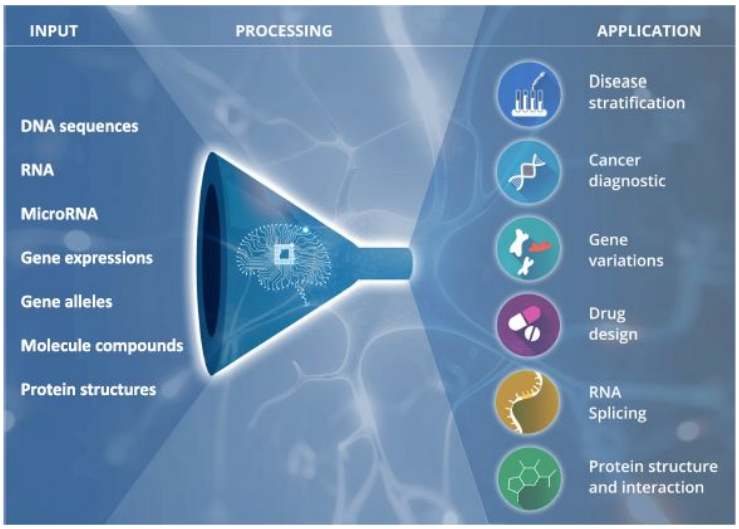


Figure 13. Overview of different inputs and applications in biomedical and health informatics where statistical learning algorithms can ascertain unknown connections between biological signatures and specified therapeutic goals. [78]

At the transcription level, next-generation sequencing technologies such as single-cell RNA sequencing have provided data for quantifying changes in the presence and quantity of RNA transcripts in the cellular transcriptomes over time or across cellular groups. Patterns of differential gene expression can be identified through supervised and unsupervised learning algorithms. Learning methodologies such as the random forest classifier can be utilized to create decision trees for predicting cellular phenotype based on the relative contribution of associated

genes [75]. Time-varying contributions of expression control mechanisms impacting cellular fates can be captured through feature representation learning and functional enrichment analysis, such as in Liang et al's development of an F-score algorithm for global expression profiles in dividing cells [80]. In pioneering understanding patterns of cellular control of neurological cells, such applications as the characterization of brain cell types through the Allen Institute for Brain Science has displayed crucial insights into the minute differences in cell types involved in neurodegenerative disorders through clustering algorithms of single-cell RNA sequencing data [81].

Attempting to capture experimentally determined and potentially unknown aspects of amino acid inputs, various encoding methodologies have been proposed for relaying information relevant to protein folding outcomes into statistical architectures (Fig. 14). Binary encoding methods utilize high-dimensional sparse vector representations, from 20 dimensional categorical one-hot encoding to lower dimensional 5-bit encodings of each amino acid, create a vectorized form in which an algorithm can recognize the distinct inputs and establish weighted calculations for optimizing a cost function [85]. Physicochemical properties based on the side chain group of each amino acid, such as hydrophobic, steric, and electric properties, can also encode targeted aspects of proposed sequences to attribute multidimensional patterns of covariation among the input sequence [85]. Position-dependent methods can generate substitution probabilities at each sequence position to generate multiple sequence alignments for a target protein sequence [85]. Furthermore, structure-based encodings can account for distinct interactions, from short-range to long-range, to reflect inter-residue contact energies [85].

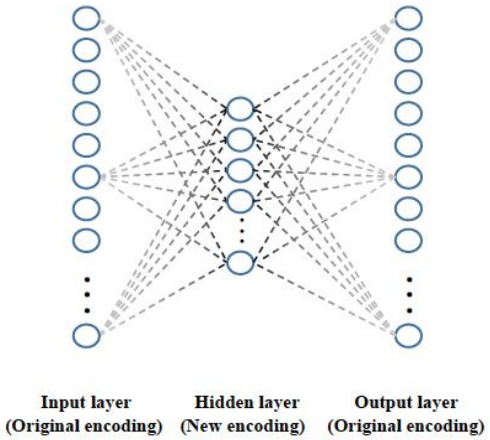


Figure 14. Generalized architecture for machine-learning based amino acid encoding methods. The input is the original sequential amino acid encodings, the output is the desired experimental measure of fitness relating to sequence, and the hidden layer represents the new numerical encodings of corresponding amino acids. [85]

Implementation of encoded amino acid sequences has shown potential in relating characteristics of natural proteins to *de novo* design. Fukuchi et al's binary classification of structural domains into either assigned domains or intrinsically disordered domains trained through sequence position-specific scoring matrices has targeted regions of unknown structural identity across a proteome [82]. Quantitative structure activity relationship analysis has presented the application of convolutional neural networks in virtual screening for protein-protein interactions in drug discovery [78]. Such binding affinity goals have produced high-throughput platforms, such as Younger el al's AlphaSeq platform for rapid characterization of synthetic sexual agglutination in reprogrammed *S. cerevisiae* cells expressing recombinant designs, that exponentially increase screening speed of targeted designs [83]. Deep residual network training of inter-residue contacts and distances based on coevolutionary data, such as the transformed restrained Rosetta method from Yang et al, extends an efficient screening process for structural goals and connects it to an energy-minimization protocol for generating structures guided by these restraints [84]. Beyond minute additions to protein domains found in nature,

deep learning has expanded the search space for novel secondary structural elements validated through parametric restraints, such as coiled-coiled assemblies and helical bundles [79].

1.7 Purpose

Among the advancements in protein stability prediction and biological systems for generating designed sequences, the aspect of expressibility is a crucial component in developing high throughput procedures for optimizing specified protein products. While the targeted application of the de novo designs, such as targeted drug binding efficiency or increased enzymatic activity, represents the ultimate goal of protein engineering, the process to achieving a successful design requires repetitive and time-consuming testing procedures to filter the massive search space for effective sequences. The ability to express these designs in a biological platform can hinder the search for effective therapies and presents a roadblock in developing validated systems for theoretical designs. As such, the goal of this work is to integrate software-based machine learning principles in expression level data of recombinant experiments in yeast in order to develop predictive measures of success extending from a given sequence of amino acids. The patterns of expressibility among various designs would allow for the screening of potential designs for their viability in binding experiments and structural determination, while also offering the generation of new potential expressing sequences extending from those showing successful integration in the yeast expression pathway.

CHAPTER 2: METHODS

2.1 Sequence Generation

In order to obtain a diverse collection of protein sequences for analyzing the collective impact of each amino acid on expression levels, approximately 100,000 sequences designed by laboratory researchers in the University of Washington's Institute for Protein Design were collected along with approximately 100,000 sequences from Two Six Labs, generated through a trained deep learning generator. These expression observations were made available through the Texas Advanced Computing Center (TACC) database storing relevant files for the DARPA Synergistic Discovery and Design (SD2) program. The datasets originated from designs resulting form the Rosetta software suite including algorithms for computational modeling and analysis of protein structures. Protein design was performed in three stages in the work of Rocklin et al: backbone construction, sequence design, and selection of designs for testing [60]. Backbone construction consisted of the *de novo* construction of compact, three-dimensional backbones with pre-specified secondary structures. Blueprint files were built for each topology in order to define a secondary structure at each residue position and the strand pairing of any β-sheets based on a set of rules relating secondary structure patterns to protein tertiary motifs. The rules were developed based on chirality orientations for three classes of junctions between adjacent secondary structure elements: $\beta\beta$, $\beta\alpha$, and $\alpha\beta$ [86]. The blueprint files were used to select three-dimensional fragments from protein crystal structures matching the proposed secondary structure. These fragments were assembled into a full protein backbone using Monte Carlo sampling with a scoring function such that the backbone matched the specified secondary

structure, satisfied compactness criteria, and avoided steric clashes. The scoring function was based on Bayesian separation of total energy into components that describe the fitness of a sequence given a particular structure [88].

The backbone structures produced were used as input for the Rosetta sequence design protocol FastDesign [89]. This protocol uses an approach called packing, where random substitutions are made using sidechain rotamers to find the sequence with the lowest possible energy for each backbone. The search for the lowest energy state is accomplished through repeated alternating rounds of rotamer optimization and gradient-based energy minimization based on amino acid repulsion. Rosetta energy functions incorporating linear combinations of terms modeling interactions between atoms, solvation effects, and torsion energies were utilized to obtain the lowest possible energy for each backbone [90]. The allowed residues at each position were restricted according to the Rosetta LayerDesign protocol, in which the environment for each residue was classified into core, boundary, or surface residues based on the solvent-accessible surface area of main-chain atoms [86]. Before the final selection stage, the designs were filtered according to compactness and overall Rosetta score [60].

The filtered designs were ranked according to several metrics, ranging from simple sequence properties such as number of hydrophobic residues to combinations of Rosetta energy terms such as full-atom energy and hydrogen bonds per residue. Multiple rounds of ranking were instituted with additional metrics such as geometric similarity to fragments of natural proteins and hydrophobic clusters. The final round involved an automated ranking scheme with structura metrics using topology-specific linear regression, logistic regression, and gradient boosting regressions to predict experimental outcome [60]. The resulting sequences were

padded with the repeating sequence GSS at the C-terminus until each sequence had a total length of 65 residues. Combined, the protein design protocol resulted in a collection of over 100,000 sequences within 22 topology classes and their associated features.

2.2 Yeast Expression Counts

The expression of designed sequences was performed through the methods described by Rocklin et al in the University of Washington's Institute for Protein Design [60]. All sequences were reverse translated and optimized for codon transcription using DNAworks2.0. This methodology optimizes the coding sequence for expression systems to avoid potential problems such as high glycine and cysteine content, codon bias, and complex intron/exon structures [91]. The sequence libraries were amplified for yeast transformation in two polymerase chain reaction steps. In the first step, 10 ng quantity, from CustomArray libraries, and a 2.5 ng quantity, from Twist and Agilent libraries, of synthetic DNA were amplified using Kapa Biosystems polymerase for 10-20 cycles by qPCR. The number of cycles was chosen to terminate the reaction at 50% yield to avoid overamplification. The reaction products were isolated through agarose gel electrophoresis and then re-amplified by qPCR. The second PCR product was purified and concentrated to remove DNA primers, nucleotide triphosphates, salts, and enzymes according to the protocol from Banauil et al [93]. A modified version of the pETcon yeast surface display vector to enable homologous recombination with designed sequences containing 40 base pairs on either end [94]. DNA libraries for deep sequencing were prepared in a similar

manner, except the amplification was started from yeast plasmid prepared from cells by Zymoprep. Six base pair barcodes were added in the second qPCR step.

S. cerevisiae yeast cells were grown and induced in preparation for cell density measurement by NanoDrop. Approximately 12-15 million cells in 1 mL increments were added to each microcentrifuge tube. The cells were washed and resuspended in a buffer solution before being labeled with anti-c-Myc-FITC for detection using chemiluminescent reagents [92]. The cells were sorted using a Sony SH800 flow cytometer, initially gated by forward-scattering and back-scattering area to collect the main yeast cell population. These cells were then gated by forward scattering width and height separately to separate individual and dividing cells from cell clumps. After these gating steps, the cells were gated by fluorescence intensity at a level of 2,200 fluorescent units to separate displaying cells from non-displaying cells. The fraction of cells passing the fluorescence threshold and the total number of cells were collected before any proteolysis steps for each sort. The libraries were assayed at six protease concentrations over three sequential selection rounds in order to assess the EC_{50} values of each sequence, providing a normalized stability score. Furthermore, the library was identified in a sequencing run using a unique six base pair barcode, and then the reads were paired using the PEAR (paired-end read merger) program for target fragments [95]. These reads were considered counts if the read contained the NdeI cut site sequence upstream from the ordered sequence, the XhoI sequence downstream from the ordered sequence, and matched the ordered sequence at the amino acid level.

2.3 Predictive Model Designs

The dataset included each amino acid sequence in the library and its associated characteristics available through sequencing and subsequent experimental analysis. Among the expression reads for each sequence, the dataset included the naive library counts of the DNA sequence corresponding to each transformed sequence, the counts with no proteolytic exposure for each expressed design passing the fluorescent threshold, and the expression counts following each level of proteolytic exposure. Expression was indicated by the naive library counts and expression counts, normalized by the ratio between the two values, whereas the proteolytic exposure counts were only used for establishing the stability score when comparing expression to protein stability [60].

Machine learning architectures were implemented in order to encode the sequences to numerical representations and formulate predicted expression levels. Models were implemented for the objectives of classification and quantitative regression utilizing the machine learning frameworks PyTorch and Tensorflow in Python. Among various architectures considered, standalone versions or combinations of three general frameworks for extracting were implemented: linear dense layer activation, the long short-term memory (LSTM) architecture of recurrent neural networks designs, and a version convolutional neural network (CNN).

Dense layers are regular deeply connected neural layers implemented either as standalone output conversions or as part of larger neural network architectures. Each neuron of a dense layer is connected to each neuron of the previous layer, forming multilayer perceptrons (MLP). Each layer contains a weight matrix W, a bias vector b, and the activations, a, of the previous layer. The values in the indices of the following layer correspond to hidden

representations capturing the linear transformations of the previous layer's values followed by a nonlinear activation function. The nonlinear activation functions, providing the ability for the network to learn by updating weights with respect to the error, utilized in the analysis included rectified linear unit (ReLU), sigmoid, hyperbolic tangent (tanh), and softmax. Dense layer architectures were incorporated in the available vector representations of each amino acid sequence (Fig. 15).

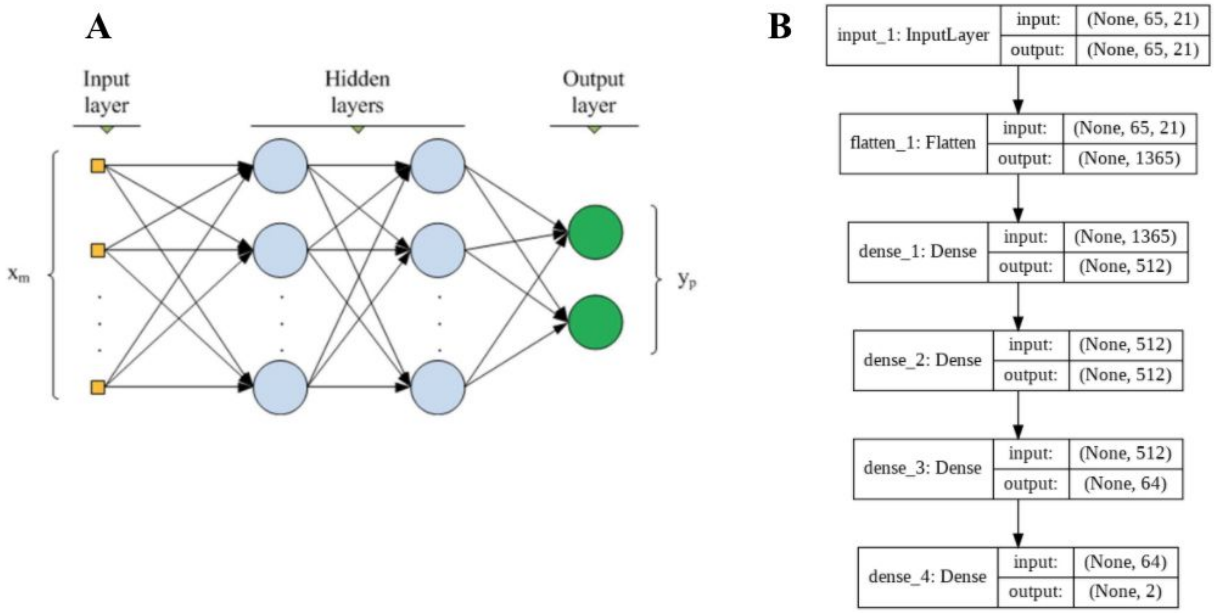
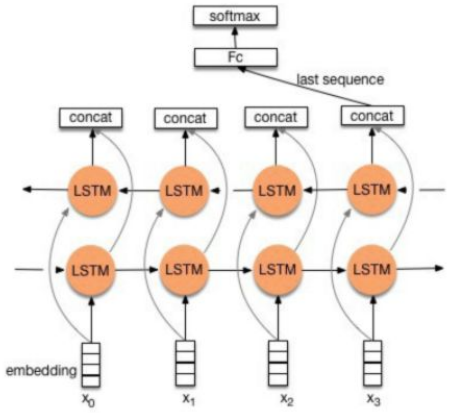


Figure 15. Model architecture for dense layer activation in Tensorflow. A) Visualization of input, hidden, and output layers. The input vectors of flattened sequence representations are connected to each neuron of subsequent layers with decreasing dimensional size. The values established in the tensor of each layer are the result of linear connections, activation functions, and regularization. B) Layer specifications in Tensorflow. [99]

Recurrent neural networks incorporate the conditional probability of an output variable at a given position given the possible effects of values in previous positions throughout a sequence. In capturing a sequence's historical information up to a current time step, each hidden state is computed recurrently with previous time steps to capture sequential effects at each position. Due to the variable impacts of other amino acids based on their distance to the position in question,

the implementation of a long short-term memory modification attempts to balance the impacts of long-term and short-term information preservation. Each hidden state contains a forget gate and a memory gate that contribute to the magnitude with which the output of each position is computed based on the learned contributions of previous positions. Because the relative contributions of each amino acid is not limited to those only seen in previous positions within a linear sequence, the bi-direction LSTM architecture was adopted to incorporate both forward and backward hidden state updates in training (Fig. 16).

A



B

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 65)]	0
embedding (Embedding)	(None, 65, 128)	2688
bidirectional (Bidirectional	(None, 128)	98816
dropout (Dropout)	(None, 128)	0
dense (Dense)	(None, 1000)	129000
dense_1 (Dense)	(None, 100)	100100
dense_2 (Dense)	(None, 2)	202

Figure 16. Bidirectional LSTM network architecture. A) Visualization of information flow for embedded inputs. B) Layer specifications in Tensorflow. [99]

Convolutional neural networks utilizes the concept of spatial invariance, such that the network responds similarly to observed regions of a sequence, and aggregates these local representations to make predictions at the entire sequence level. Rather than having a single activation corresponding to each spatial location, convolutions produce entire vectors of hidden representations corresponding to each spatial location. These feature maps provide a spatialized set of learned features passed on to the subsequent layer. In encoded sequences, this operation

amounts to a two-dimensional cross-correlation of a kernel operator passed over the input matrices and the addition of a bias term. This operation is followed with a pooling layer to reduce the spatial size of each convolved feature. The convolution layers in this architecture were dilated to capture potential longer range impacts of neighboring residues. The convolutional network implemented for analysis incorporated a modified version of the ResNet architecture in which repetitions of convolutional and pooling layers are added to the initial input to help mitigate the problem of vanishing gradients (Fig. 17, Fig. 18).

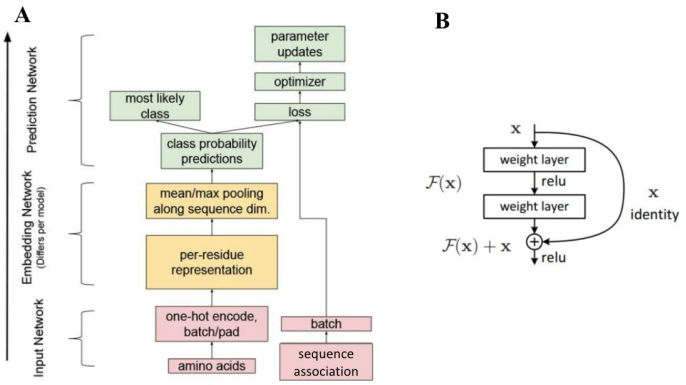


Figure 17. Convolutional network representation. A) The information flows through the convolutional network, incorporating encoded amino acid sequences and associated measures of performance. B) The skip connection adds the input information for the convolutional block to the output of convolutional operations. [99]

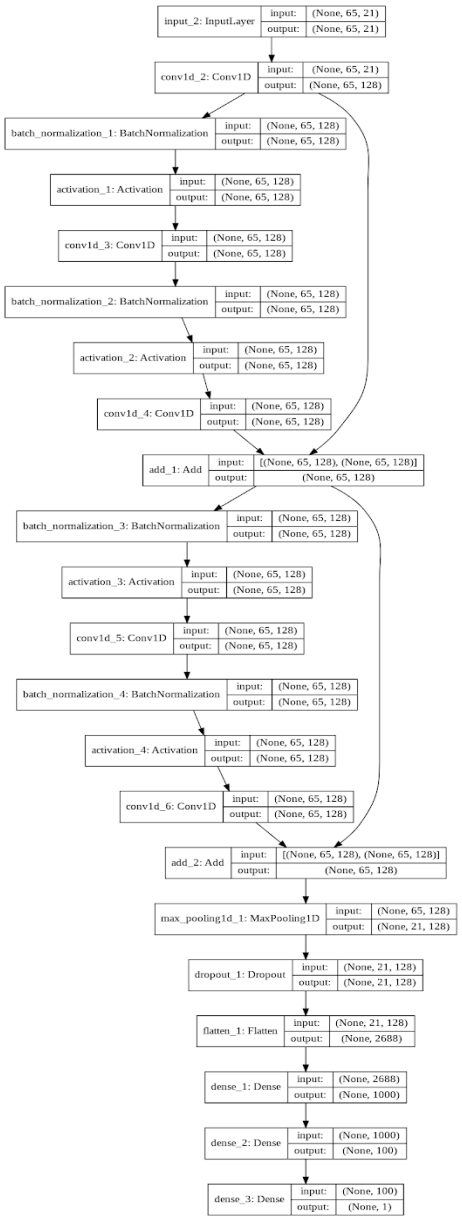


Figure 18. Model architecture of the convolutional neural network in Tensorflow.

The architectures of the models were adjusted across multiple iterations to find those with the best validation accuracy in their respective predictive goals. Those with the best achieved predictive performance in each framework were reported. For classification models, the loss function of categorical cross-entropy was used, which accounts for the product of the ground truth and the predicted output class score for each sequence. For quantitative prediction models,

the loss function used was mean square error, which accounts for the average square of the difference between the predicted outcomes and the true values for each encoded sequence. Very little difference in results based on the choice of activation functions was observed, so rectified linear unit (ReLU) was used for connected layers other than those in the final output layers. The final layers either used a softmax activation for classification or a linear activation for quantitative outputs.

2.4 Encoding Methods

Several encoding methodologies were tested in representing the incoming sequences in numerical forms which could be analyzed (Fig. 19). The primary method for analysis was the one-hot encoding of each amino acid. In this methodology, each amino acid is represented by a vector of length 21 to account for each of the 20 amino acids and an additional value for any miscellaneous values. Within each vector, all values are zero except for the index corresponding to the amino acid, which received a value of one. Each sequence was thus represented by a 65x21 matrix of concatenated one-hot encoded vectors. To analyze performance based on amino acid hydropathy, a matrix consisting of hydropathy differences between residues at each position was also created for each sequence. In each matrix, the value at each index is the difference between the hydropathy values of two amino acids, one in the position corresponding to the row and the other in the position corresponding to the column. Thus, each sequence in this encoding was represented by a 65x65 matrix with the difference in hydropathy values between two amino

acids in each entry. Finally, based on encoding methods in natural language processing, each amino acid type was also embedded in a dense vector representation based on a dictionary of the 20 amino acid codes. This embedding conversion starts with an initialized set of random weights and learns a numerical vectorized embedding of specified length for each amino acid code. Thus, the amino acid letters of each sequence were embedded into a 65x128 matrix of numerical embedded values.

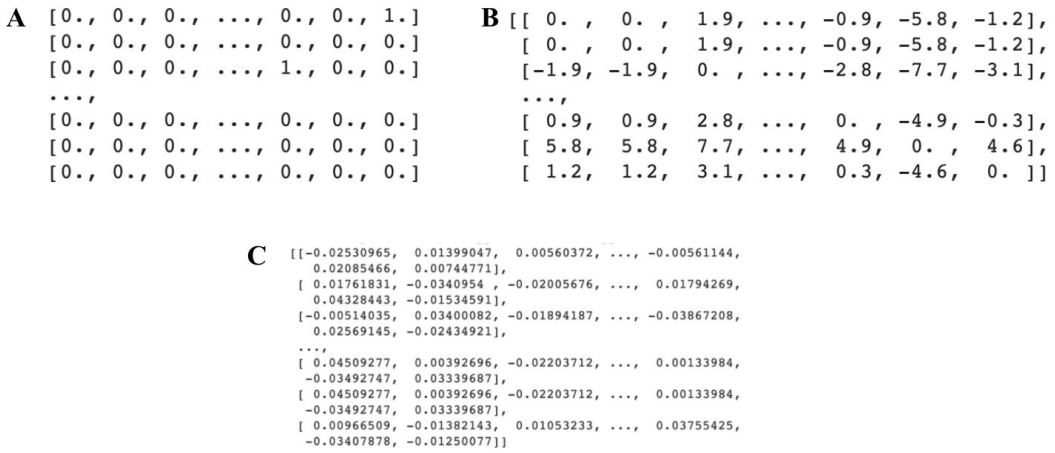


Figure 19. Encoded matrices of amino acid sequences. A) One-hot encoding matrix (65x21). B) Hydropathy difference matrix (65x65). C) Language embedding matrix (65x128).

These encoded sequences were fed through the network for learning predicted outcomes based on experimental fitness. The first measure of expressibility was established as a binary classification in which each sequence would be considered to either be enriched or depleted in cell sorting the fluorescent expressed sequences when compared to the amount in the naive library. The cutoff between the two classes was set based on the ratio of expressed counts over naive counts at a value of one. The second measure of expressibility was the numeric value of this ratio between expressed counts and naive counts. Finally, the expressed counts alone were

used as the measure of expressibility. Due to the uncertain nature of the variability in the amount of each sequence transformed in yeast, the differences in predictive performance on these values were compared. The sequences were split into training (80%) and validation (20%) sets for analyzing model performance (Fig. 20). Within the aggregated dataset of Rosetta-designed structures, the models were also trained using the categorical output of topology and the numerical output of stability score.

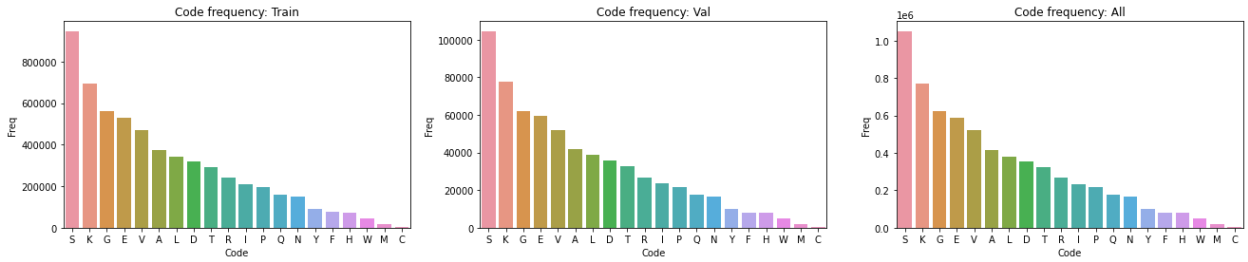


Figure 20. Amino acid code frequencies for training, validation, and total dataset.

2.5 Latent Space Exploration

In order to generate possible design alterations that could achieve improved expression performance, a variational autoencoder (VAE) was implemented to explore the latent space representations of the sequences. This VAE architecture works to create a bottleneck for input data that ensures only the most relevant factors determining an observation's structure can pass before being reconstructed to the original dimensions of the input data (Fig. 21). This architecture is based on examples in handwriting recognition and language translation where input representations are converted to two-dimensional representations in latent space. This latent space is sampled and translated to the original dimensions of the input. This model

architecture offers both training in the accurate lower-dimensional representation of complicated sequences and in the generation of new sequences from this latent space. This hidden representation was analyzed for comparisons among altered sequences and utilized to generate new sequences

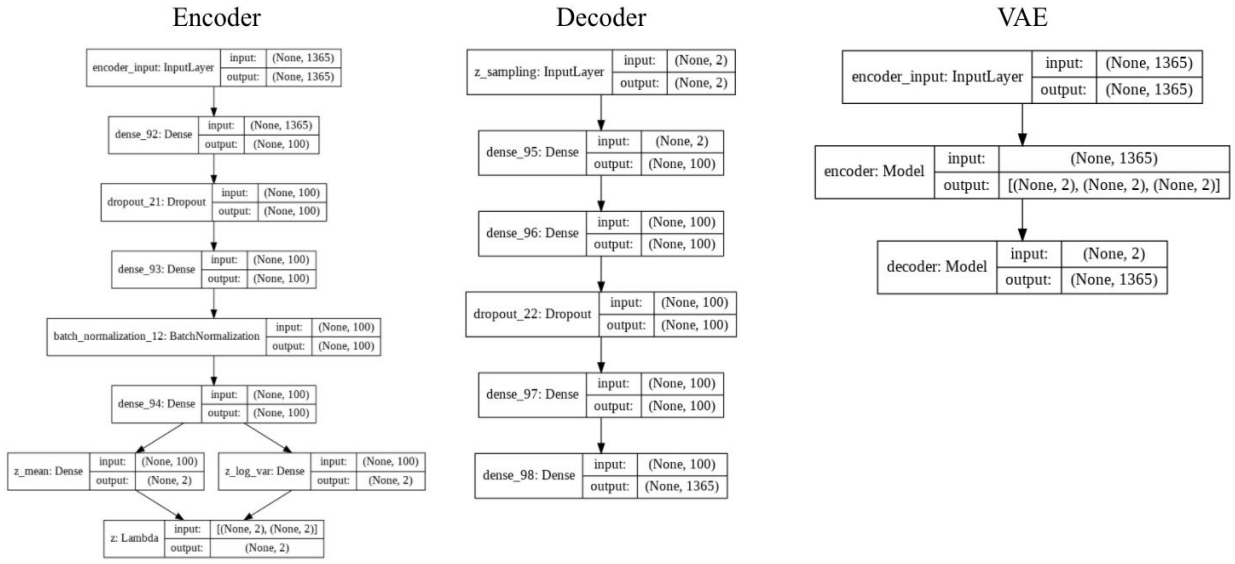


Figure 21. Architecture of variational autoencoder. The encoder compresses the input data down to two latent variable representations. The decoder samples from the latent space and transforms them back to the original dimensions of the input data. Together, these two architectures form the overall structure of the variational autoencoder (VAE).

CHAPTER 3: RESULTS

3.1 Predictive Accuracy

The binary classification of designed sequences into those with increased expression was made possible with varying degrees of success. The validation accuracy of binary predictions ranged from 67.4% to 76.3%. The dense layer activations tended to show the least predictive accuracy in the validation sets, whereas the convolutional neural network architecture had the highest predictive accuracy. Interestingly, dense layer activation was able to achieve a high training accuracy/low training error estimate with both outputs, yet its validation Furthermore, the most successful encoding method appeared to be the categorical one-hot encoding of each amino acid when compared to the other two methods. The same differences in performance were observed in the prediction of the continuous variable of expression ratio. The convolutional neural network displayed the lowest mean squared error (MSE) of 0.412 in the validation set using one-hot encoding of the input amino acids.

A

Binary Classification			
Model	**Inputs**	**Train Accuracy**	**Val. Accuracy**
1: Dense Layers	1hot encoding	91.2%	67.4%
2: Dense Layers	Hydropathy difference	90.6%	66.8%
3: Dense Layers	Embedding layer	86.7%	66.9%
4: Bi- LSTM	1hot encoding	92.2%	71.2%
5: Bi-LSTM	Hydropathy difference	93.1%	71.8%
6: Bi-LSTM	Embedding layer	87.1%	72.1%
7: CNN	1hot encoding	93.1%	76.3%
8: CNN	Hydropathy difference	94.2%	73.1%
9: CNN	Embedding layer	89.2%	74.1%

B

Expression Ratio Prediction			
Model	**Inputs**	**Train MSE**	**Val. MSE**
1: Dense Layers	1hot encoding	0.196	0.598
2: Dense Layers	Hydropathy difference	0.274	0.613
3: Dense Layers	Embedding layer	0.341	0.625
4: Bi- LSTM	1hot encoding	0.542	0.548
5: Bi-LSTM	Hydropathy difference	0.589	0.596
6: Bi-LSTM	Embedding layer	0.596	0.532
7: CNN	1hot encoding	0.294	0.412
8: CNN	Hydropathy difference	0.412	0.541
9: CNN	Embedding layer	0.512	0.591

Table 3. Accuracy of different model architectures and encoding methods for target outputs. A) Binary classifier for predicting whether a sequence will be enriched in expression counts. B) Continuous variable prediction for ratio of expression to naive counts.

Through the binary classifier, the relative probability of a sequence being enriched in the group of expressed counts compared to the group of naive counts allowed for a relative comparison of expressibility across all sequences (Fig. 22). The model was trained on the experimental data from the deep learning model generated designs, and it was applied to the designs from the Institute for Protein Design to visualize the distinctions among the labeled topologies. In general, the probability of enrichment among expressed proteins was most similar among topologies

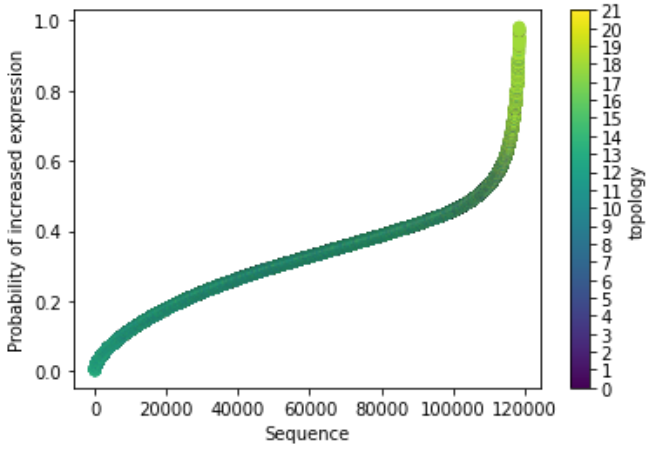


Figure 22. Ranked predicted probability of sequence enrichment among expressing proteins, colored by topology class.

The prediction of the continuous output of expression ratio, defined as the ratio of expressed counts to naive counts, produced values that tended to increase along with an increase in their true values (Fig. 23). While some designs showed larger deviation from their expected values, the increase in predicted values showed a loosely linear increase as the true value of the expression ratio increased. The coefficient of determination (R^2) value between the true expression ratio and predicted expression ratio was 0.672, suggesting a trend in increased predicted expression ratio along with an increase in the true expression ratio.

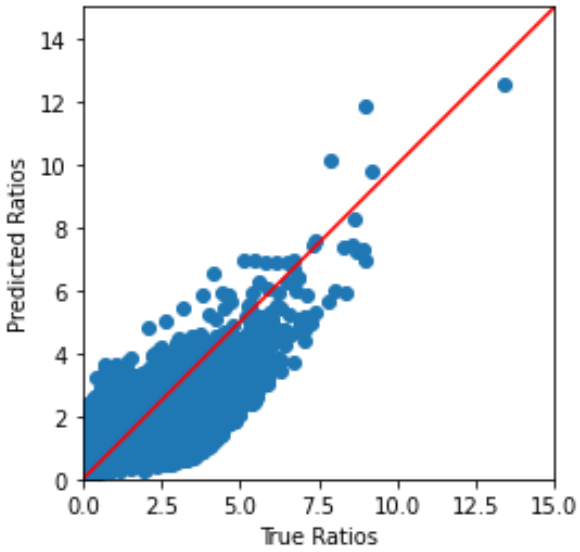


Figure 23. Plot of predicted expression ratio (ratio of expressed counts to naive library counts) through CNN compared to true values. The red line indicates where the two values are equal to one another.

The same model architectures were applied to the topology classes and stability scores associated with each sequence (Table 4). As these features have been experimentally determined in other experiments, they offer a basis of comparison both for the efficacy of this model as well as associations of expression data with these features. The predictive models performed very well on extracting topology classes from the provided sequences, with the convolutional neural network classifying the topology of each sequence with 99.7% accuracy in the validation set. The models also performed reasonably well with the stability score as the output, with the CNN architecture displaying the lowest validation MSE of 0.260.

A

Topology Classification			
Model	**Inputs**	**Train Accuracy**	**Val. Accuracy**
1: Dense Layers	1hot encoding	94.5%	96.3%
2: Bi-LSTM	1hot encoding	91.3%	94.8%
3: CNN	1hot encoding	99.1%	99.7%

B

Stability Score Prediction			
Model	**Inputs**	**Train MSE**	**Val. MSE**
1: Dense Layers	1hot encoding	0.329	0.412
2: Bi-LSTM	1hot encoding	0.314	0.358
3: CNN	1hot encoding	0.239	0.260

Table 4. Accuracy of different model architectures for A) topology classification and B) stability score prediction.

Comparing the predicted values of sequence expression ratio to stability score, the predicted expression did not show a clear linear trend with the predicted stability score (Fig. 24). A given predicted expression level showed designs with a wide range of stability scores. When labeled by topology, several topologies tended to cluster with similar expression levels, while others showed more variation in predicted expression levels across their set of designs. Observing if there were any trends between expression ratio and stability score, it was not clear if one predicted feature had a specific impact on the other.

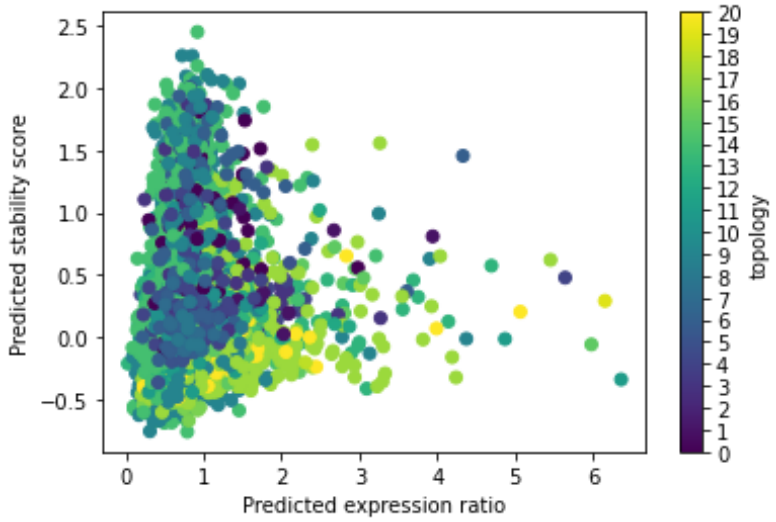


Figure 24. Comparison of predicted stability score to predicted expression ratio, colored by topology class.

3.2 Unsupervised Latent Space Representation

Sequences closely matching designed sequences were sought after for improved expression performance. The representation of the trained sequences in latent space showed the capabilities of the variational autoencoder to locate similar sequences in two-dimensional latent space (Fig. 25). When colored by topology, the latent space representation distributed the

sequences in similar locations among the 2-dimensional space. This space displays the sample space from which the decoder recreates sequences based on position-based probability predictions.

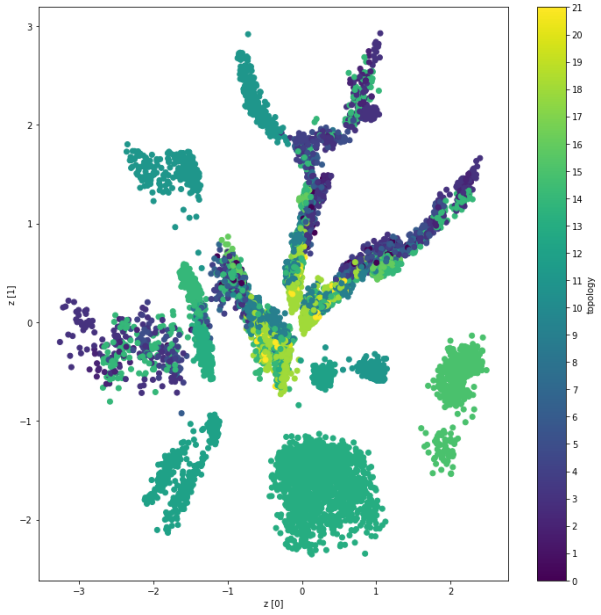


Figure 25. Latent space representation of aggregated Rosetta designs colored by topology classification.

A designed sequence was chosen as a reference for analyzing the effects of single point mutations on the representation of the sequence. Multiple sequences were generated from this reference sequence by mutating single amino acids at each location with another chosen from the collection of all natural amino acids. All designs were plotted in the two-dimensional latent space representation, and the reference sequence along with the mutated variants were separated for visualization (Fig. 26). Additionally, the sequences not part of the set of mutants were colored according to the cluster labels resulting from their partitioning through k-means clustering, which showed slightly different clusters than those seen by simply labeling according to topology classification. As seen in the latent space representation, the degree and direction of separation between the original sequence and the mutated variants differed depending on the

location and type of amino acid substitution, displaying the unique impacts of a single amino acid on the encoded nature of a polypeptide sequence.

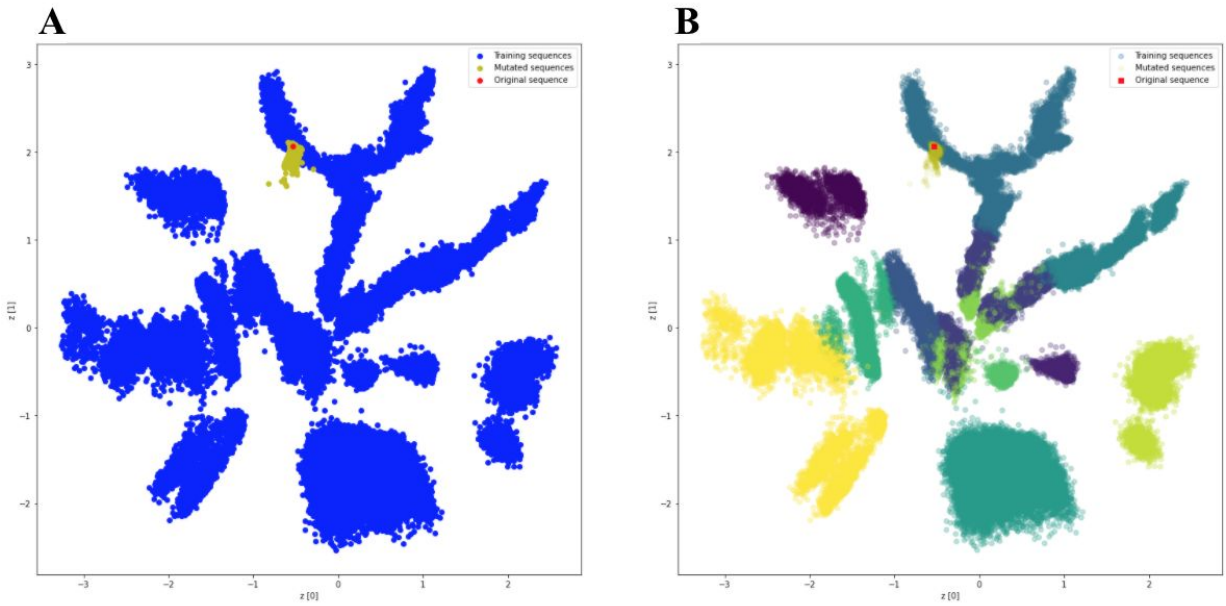


Figure 26. Latent space representation of aggregated data in variational autoencoder. A) All Rosetta designed sequences colored in blue. Observed sequence colored in red. Single mutation variants of observed sequence colored in yellow. B) Same plot with all other sequences colored according to k-means cluster labels.

Through the encoding of these variants in the latent space, their subsequent decoded outputs present distinctions in their predicted sequences. Reconstructions of latent space locations creates a probability weight matrix in which the neural network assigns a probability in among all 20 amino acid options at each position (Fig. 27). Single point mutations create changes in the output probability matrix across all positions, not just the one where the mutation occurred. This change in predicted probability displays the cascade of effects resulting from changing minute regions of a sequence. Due to the complex nature of creating a low-dimensional space of sequences composed of 65 amino acids, the direction of changes resulting from small changes in the sequence offer visual insight into how the variational autoencoder is learning to minimize the loss of decoded accuracy.

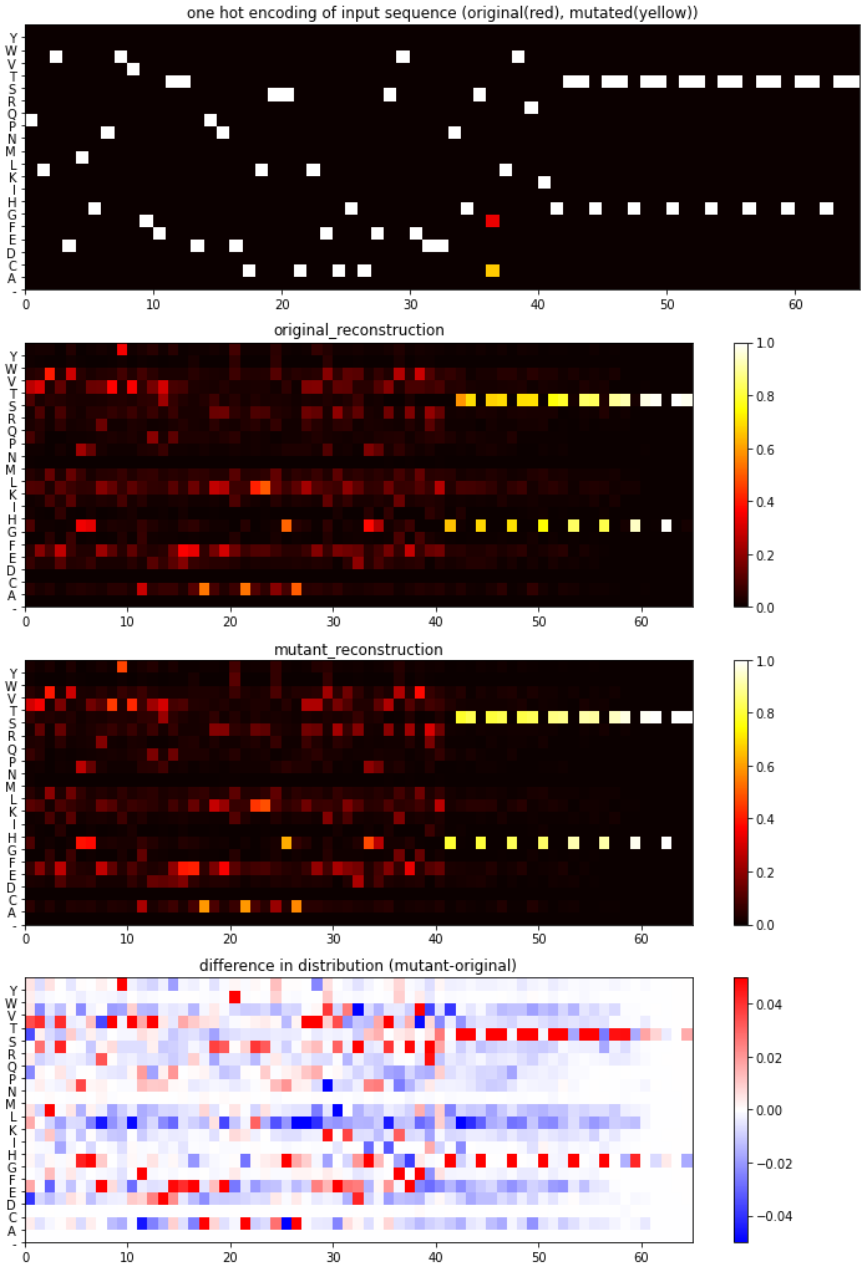


Figure 27. Reconstructed sequences from latent space represented by one-hot encoding of each residue position. The top figure represents the one-hot encoding of each position in the amino acid sequence. The positions indicating the amino acid type are the same at all positions except for the one where the mutation occurred. The position of the original amino acid type is colored in red, while the one indicating the type of the mutated sequence is yellow. The middle two plots indicate the probability weight matrix for both the reference sequence and the mutated sequence. These weight matrices were the result of decoding the latent space representations of each sequence. The bottom figure represents the difference in the values of the probability weight matrix between the reference sequence and the mutated sequence.

Principal component analysis (PCA) was applied to the set of mutated sequences to find the direction in latent space accounting for the majority of the variance among their encodings. Moving along the first principal eigenvector from PCA analysis, the alterations in latent space present new sequences that reflect the change seen from single point mutations. The goal of analyzing sequence fitness along this vector is to move in latent space in a manner that reflects small changes to a target sequence. The vectors along which the VAE minimizes representational loss in this space point to potential gradients in which the change in output topology can be analyzed, possibly creating new combinations of amino acids along an observable gradient.

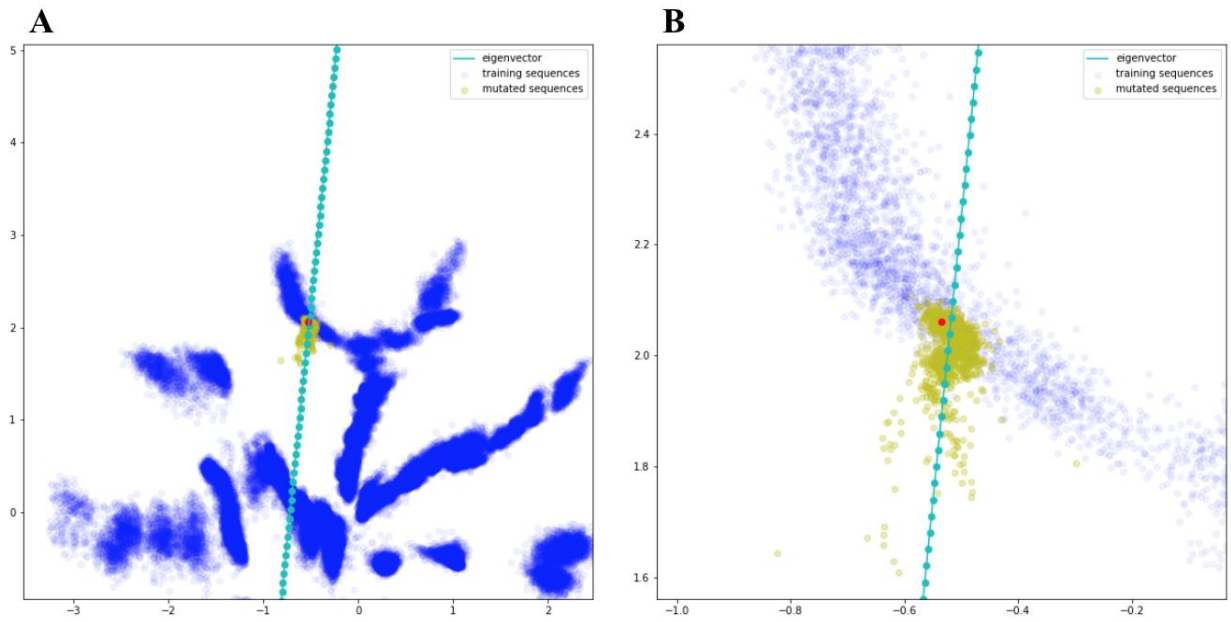


Figure 28. Plot of principal component analysis of mutated sequences on latent space representation. A) View of overall latent space plotted with PCA eigenvector (in turquoise) of mutated sequences. B) Zoomed in view of mutated sequences. Each dot along the eigenvector represents a small perturbation in latent space to test for decoded output.

Perturbing incremental steps along the PCA eigenvector of the mutated sequences subsequently allows for the visualization of predicted sequence decoding (Fig. 29, Fig. 30). Moving along the first principal component vector, corresponding to the direction that captures the most variance in the latent space reconstruction of the mutated sequences, the decoded output sequence changes significantly, particularly when moving further away from the locations in which the training sequences were observed. The subsequent changes in decoded sequences offers the generation of sequences not used in training and how they relate to the variation in sequences resulting in different expression levels.

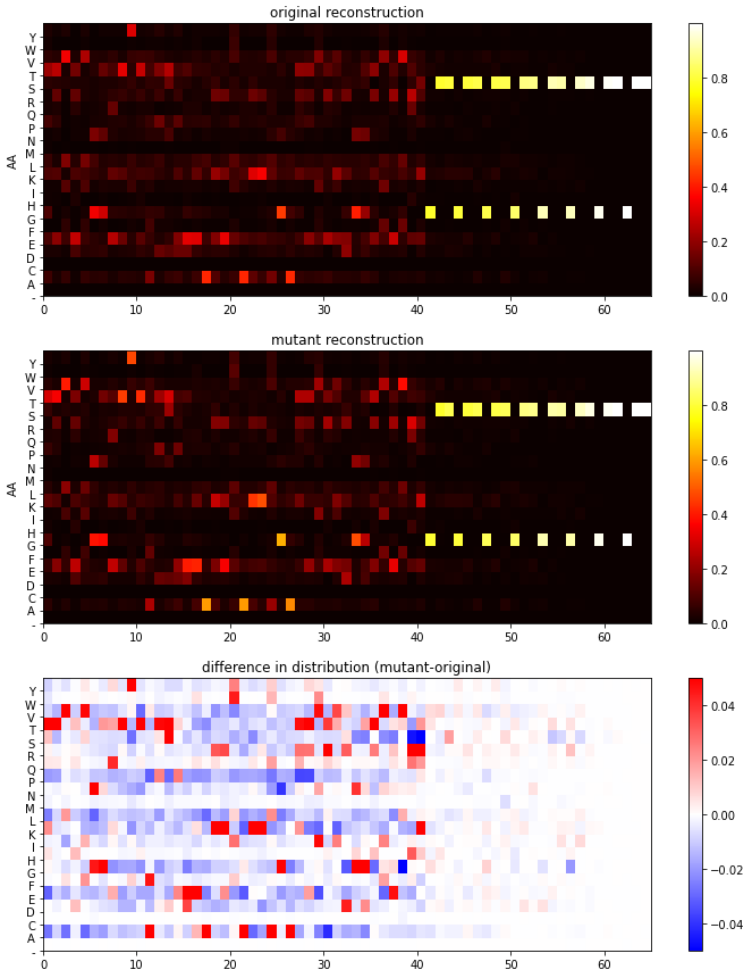


Figure 29. Reconstructed sequences from latent space representation. The top plot displays the original sequence's reconstruction. The middle plot displays the reconstruction following a perturbation in latent space along the eigenvector of PCA analysis. The bottom plot displays the difference in reconstructed predictions between the reconstructed original sequence and the perturbed reconstruction.

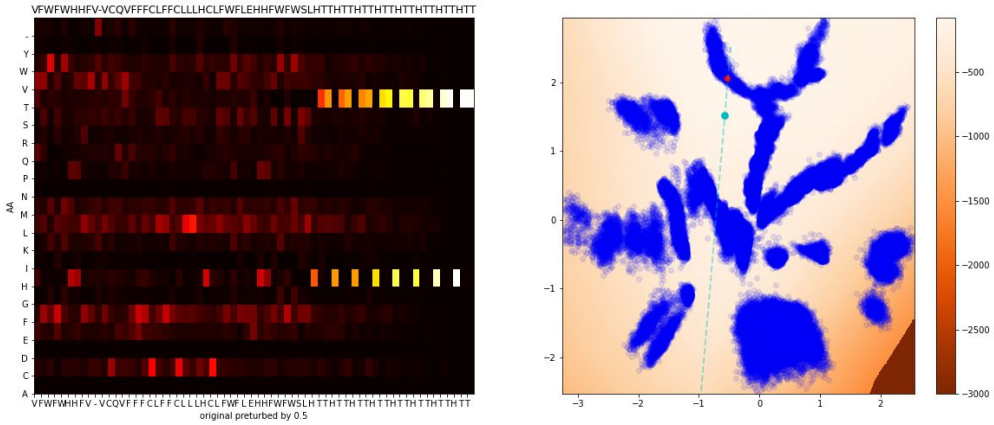


Figure 30. The left shows the sequence reconstruction based on the location within the latent space of the trained variational autoencoder. Each perturbation corresponds with a position-weight matrix used to predict the most likely sequence given the coordinates in latent space. The right shows the movement within the latent space representation. The color of the background corresponds to the log probability of decoding the original reference sequence within the latent space representation.

3.3 Trained Model Dimensionality Reduction

After training the models with the supervised outputs of expression metrics, the sequences were passed through the trained architecture in order to visualize the distribution of values (Fig. 31). These vectors of length 100 were compressed into two latent variable representations similarly to the hidden state of the variational autoencoder. The two-dimensional latent space shows the distribution of predicted expression ratios based on the one-hot encoding of their amino acid sequences. The plot displays a general tiered segmentation of the values, tending to increase as the values of both latent variables increased (diagonally top and to the right). The plot of the sequence and its single-point mutations display how each sequence representation impacts its location in latent space and the subsequent predicted expression

metric. Mutating the sequences tended to move their hidden representations along a diagonal from top left toward bottom right in the plot, corresponding to close expression metrics (as indicated by the color gradient of all the points). Because these sequences were passed through a separate model trained only on the output metric of expression, the two-dimensional representation is, as expected, different than that based on the variational autoencoder, which was trained on the decoded output of the representations.

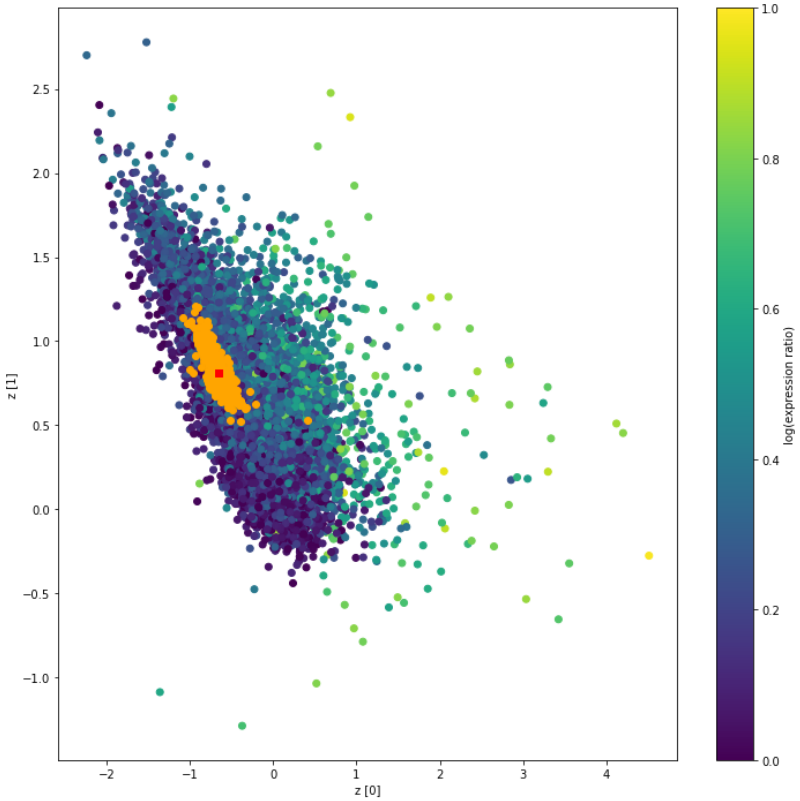


Figure 31. Two-dimensional latent space representation of the final layer of the trained convolutional neural network. The reference sequence is colored in red, the sequences of single mutations are colored in orange, and the remaining sequences are colored according to the values of their predicted expression ratios as output from the CNN.

3.4 Sequence Expression Comparison

The reference sequence and the single-residue mutant variants visualized in the VAE were compared by passing their one-hot encodings through the trained classification model (Fig. 32). Each variant produced a different probability of enrichment among expressing proteins. There were multiple mutated variants that either decreased or increased the binary probability compared to the original reference sequence. The supervised learning method previously described presents an ordering of the most preferred single mutations in a target sequence in terms of expressibility, presenting possible iterative changes to a desired sequence outcome that could improve expression in a surface display assay.

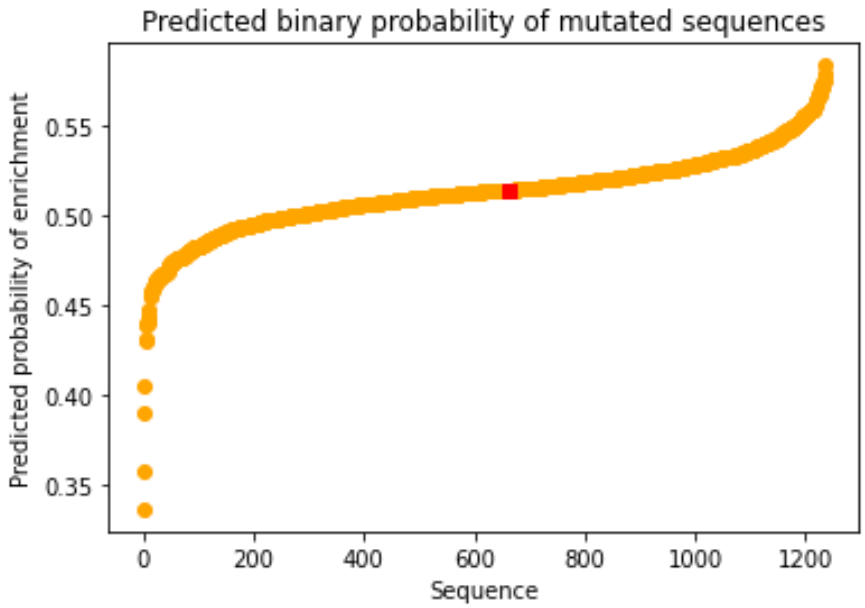


Figure 32. The impact of single-residue mutations on predicted expression enrichment. The plot shows the probabilities from the trained CNN binary classifier, with the reference sequence colored in red and all sequences with single-residue mutations colored in orange.

The variational autoencoder's latent space itself was also used to generate new sequences not observed in the training data. The orientation of the two-dimensional space ideally presents similar sequences, as determined by training the decoded outputs of the space, in close proximity

to one another in the space (Fig. 33). Making small perturbations in this space was used to generate 400 sequences close (within a random range within 0.1 units of both latent space parameters) to the top 10 most highly expressed sequences within each of the 22 given topologies in the training data. These sequences were then fed into the trained expression network to predict their expression levels (Fig. 34). The comparison of sequences similar to the top 10 most expressed sequences in the topology "0a6b" are displayed for visualization. While many of the sequences did not appear to show a high expression prediction (as displayed in their predicted binary classification probabilities and their predicted expression ratios), several sequences showed promising expression predictions that could match or even exceed the expression of those seen in the surface display assay. These sequences offer potential alternative designs that could create a more successful expression trial with targeted designs similar to those of interest in the original protocol.

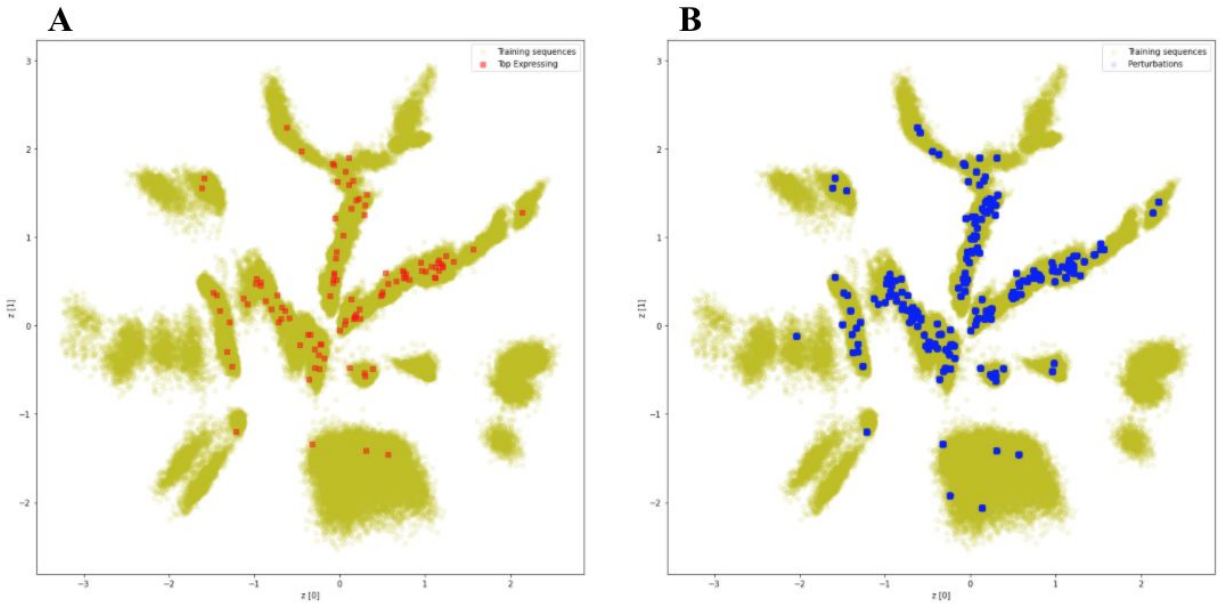


Figure 33. A) Locations of top 10 most highly expressed sequences for each of the 22 provided topologies represented in the VAE's latent space shown in red. B) Small perturbations away from the locations of the most highly expressing proteins, 400 for each topology, in latent space shown in blue.

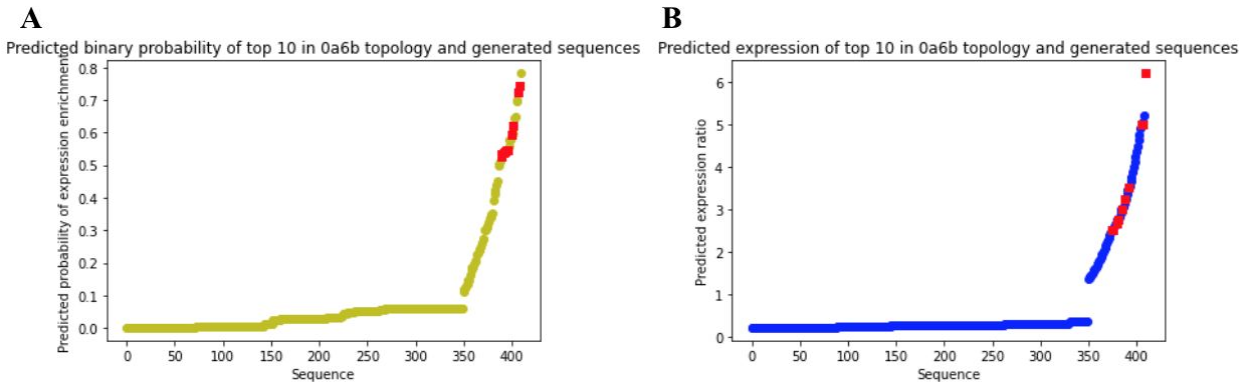


Figure 34. The predicted expression metrics of the top 10 expressed sequences in the class "0a6b." In both plots, the top 10 most highly expressing sequences are colored in red. The other sequence predictions are from decoded perturbations in the variational autoencoder's latent space. A) The predicted binary probability of an increased presentation among expressing proteins for the sequence reconstructions. B) The predicted expression ratio of expressing counts to naive counts for the sequence reconstructions.

CHAPTER 4: DISCUSSION

The processes of this study presented a use case for machine learning principles in expression of both observed and newly generated sequences in a yeast surface display assay. While many studies have been conducted in the application of *de novo* designs to their predicted stability and structure, this study addressed a gap in analyzing the crucial aspect of expressing these sequences for efficient analysis of potentially successful folding outcomes. Machine learning principles showed a degree of success, measured by their errors in predicted outcome classifications and numerical predictions, in being able to gather patterns from complex high dimensional sequences and relate those to the target of expression. Similarly, the reduction of these higher dimensions was made possible through machine learning architectures to both visualize the distinctions among these various combinations of amino acids and generate new sequences with similar representations to those seen in training data.

While this study analyzed the sequences of one assay, its application could be improved and extended to other applications with repeated experiments. One aspect of this study to address is the differences in the naive counts of each sequence observed in the sample. While this value was used to normalize the expression counts in creating an expression ratio for the target output, it would be useful to know if that variation remains constant in repeated experiments. Knowing whether that variation is due just to random sampling noise in a set of billions of cells or if it is a relevant metric of how well these sequences can be transformed into the cells would help to refine the model and determine if that aspect of recombinant genetics ultimately impacts the expression levels in addition to the sequence's pathway from the genome to cell surface expression. Furthermore, increasing the diversity of the training data could help

to refine patterns that are learned in the models in making predictions. This study was based on a library of approximately 100,000 sequences among 22 topologies. The more diversity in sequences and associated topologies could provide more clear distinctions in amino acid combinations that ultimately impact expression and make the model's separation of sequences based on the

In addition, validation of the model with subsequent expression assays is needed to verify if the networks are in fact making biologically accurate predictions based on these sequences. Repeating the experiment with small alterations to the library would provide insight into the small impacts of mutations in expression and whether these noticed differences are consistent across different trials. For the generated sequences extending from decoding representations from the variational autoencoder, it would also be necessary to validate that these sequences, which achieved a predicted expression level through these models, do in fact show similar expression levels when transformed in a yeast library. The model made predictions based on the sequences seen in the training library, yet the ultimate use of this platform would be to ensure that these targeted generations can improve upon the expression of other sequences.

The broader value of this procedure in protein engineering would come from its extension into targeted protein structure folding outcomes. It has shown promise in filtering the search space for designs intended for a specific binding and therapeutic function, but integrating it into an iterative workflow of protein design, expression, and binding affinity would help to both refine the model's outcomes and produce tangible results in terms of finding designs with the highest probability of fulfilling their intended purpose. For instance, the processing of potential designs for the binding analysis of designs, such as in the AlphaSeq platform of

A-Alpha Bio, provide a direct extension of progressing targeted designs through to the testing stage of therapeutic development. Furthermore, this platform could be used in conjunction with biophysical models to help refine their chemical and electrostatic projections of the atomic structure of each sequence. As an accurate theoretical representation of these sequences is necessary for developing novel structural elements, this expression prediction could help to further screen sequences in the design process and emphasize the ability for these desired targets to be expressed in a biological setting. Additionally, extending this analysis to other transgenic applications could help to elucidate the success of gene therapy in eukaryotic cells. Adjacent to the goal of predicting expression in biological platforms, similar analysis could be done for the successful integration of gene's in live mammalian cells, presenting the successful translation of proteins in a dynamic cellular environment aiding in a therapeutic outcome.

REFERENCES

1. Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O., & Eisenberg, D. (1999). Detecting protein function and protein-protein interactions from genome sequences. *Science*, *285*(5428), 751–753. https://doi.org/10.1126/science.285.5428.751
2. Lee, D., Redfern, O., & Orengo, C. (2007). Predicting protein function from sequence and structure. *Nature Reviews Molecular Cell Biology*, *8*(12), 995–1005. https://doi.org/10.1038/nrm2281
3. Chevalier, A., Silva, D. A., Rocklin, G. J., Hicks, D. R., Vergara, R., Murapa, P., … Baker, D. (2017). Massively parallel de novo protein design for targeted therapeutics. *Nature*, *550*(7674), 74–79. https://doi.org/10.1038/nature23912
4. Woolfson, D. N., Bartlett, G. J., Burton, A. J., Heal, J. W., Niitsu, A., Thomson, A. R., & Wood, C. W. (2015). De novo protein design: How do we expand into the universe of possible protein structures? *Current Opinion in Structural Biology*, *33*, 16–26. https://doi.org/10.1016/j.sbi.2015.05.009
5. Li, Z., Yang, Y., Zhan, J., Dai, L., & Zhou, Y. (2013). *Energy Functions in De Novo Protein Design : Current Challenges and Future Prospects*. https://doi.org/10.1146/annurev-biophys-083012-130315
6. Genchi, G. (2017). An overview on d-amino acids. *Amino Acids*, *49*(9), 1521–1533. https://doi.org/10.1007/s00726-017-2459-5
7. Błażewicz, J., Łukasiak, P., & Miłostan, M. (2006). Some operations research methods for analyzing protein sequences and structures. *4or*, *4*(2), 91–123. https://doi.org/10.1007/s10288-006-0089-y
8. Hruby, V. J. (1982). Conformational restrictions of biologically active peptides via amino acid side chain groups. *Life Sciences*, *31*(3), 189–199. https://doi.org/10.1016/0024-3205(82)90578-1
9. Caplan, M. R., Schwartzfarb, E. M., Zhang, S., Kamm, R. D., & Lauffenburger, D. A. (2002). Control of self-assembling oligopeptide matrix formation through systematic variation of amino acid sequence. *Biomaterials*, *23*(1), 219–227. https://doi.org/10.1016/S0142-9612(01)00099-0
10. Bowie, J. U., Reidhaar-Olson, J. F., Lim, W. A., & Sauer, R. T. (1990). Deciphering the message in protein sequences: Tolerance to amino acid substitutions. *Science*, *247*(4948), 1306–1310. https://doi.org/10.1126/science.2315699
11. Neuberger, A. (2019). Stereochemistry of Amino Acids. *Advances in Protein Chemistry*, *4*(C), 297–383. https://doi.org/10.1016/S0065-3233(08)60009-1
12. Biro, J. C. (2006). Amino acid size, charge, hydropathy indices and matrices for protein structure analysis. *Theoretical Biology and Medical Modelling*, *3*(1), 1–12. https://doi.org/10.1186/1742-4682-3-15
13. Kyte, J., & Doolittle, R. F. (1982). A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, *157*(1), 105–132. https://doi.org/10.1016/0022-2836(82)90515-0
14. Diamond, R. (1971). A real-space refinement procedure for proteins. *Acta Crystallographica Section A*, *27*(5), 436–452. https://doi.org/10.1107/S0567739471000986

15. Xu, S., Nilles, J. M., & Bowen, K. H. (2003). Zwitterion formation in hydrated amino acid, dipole bound anions: How many water molecules are required? *Journal of Chemical Physics*, *119*(20), 10696–10701. https://doi.org/10.1063/1.1620501
16. Frydman, J. (2001). Folding of Newly Translated Proteins In Vivo: The Role of Molecular Chaperones. *Annual Review of Biochemistry*, *70*(1), 603–647. https://doi.org/10.1146/annurev.biochem.70.1.603
17. Tyers, M., & Mann, M. (2006). From genomics to proteomics. *Insight Overview*, *422*(March), 193–197.
18. Aasland, R., Abrams, C., Ampe, C., Ball, L.J., Bedford, M., Cesareni, G., Gimona, M., Hurley, J., & Jarchau, T. (2002). Normalization of nomenclature for peptide motifs as ligands of modular protein domains. *FEBS Letters*. *513*(1), 141–144. https://doi.org/10.1016/S0014-5793(01)03295-1
19. Xu, D., Tsai, C. J., & Nussinov, R. (1997). Hydrogen bonds and salt bridges across protein-protein interfaces. *Protein Engineering*, *10*(9), 999–1012. https://doi.org/10.1093/protein/10.9.999
20. Pentelute, B. L., Gates, Z. P., Tereshko, V., Dashnau, J. L., Vanderkooi, J. M., Kossiakoff, A. A., & Kent, S. B. H. (2009). *NIH Public Access*. *130*(30), 9695–9701. https://doi.org/10.1021/ja8013538.X-ray
21. Yang, A. S., & Honig, B. (1995). Free Energy Determinants of Secondary Structure Formation: I. A-Helices. *Journal of Molecular Biology*, *252*(3), 351–365. https://doi.org/10.1006/jmbi.1995.0502
22. Richardson, J. S., & Richardson, D. C. (2002). Natural β-sheet proteins use negative design to avoid edge-to-edge aggregation. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(5), 2754–2759. https://doi.org/10.1073/pnas.052706099
23. Kolinski, A., & Skolnick, J. (1997). Determinants of secondary structure of polypeptide chains: Interplay between short range and burial interactions. *Journal of Chemical Physics*, *107*(3), 953–964. https://doi.org/10.1063/1.474448
24. Ardejani, M., Chok, X. L., Foo, C. J., & Orner, B. (2 April 2013). Complete shift of ferritin oligomerization toward nanocage assembly via engineered protein–protein interactions. *Chemical Communications*. *49*(34): 3528–3530. https://doi.org/10.1039/C3CC40886H. ISSN 1364-548X
25. Showmy, K.S., Skariyachan, S., & Yusuf, A. (2014). Comparative modelling of pathogenesis related 4b protein (Q6T5J8) of Oryza sativa subsp. indica with the three-dimensional structure of barley1BW3. International Journal of Plant, Animal and Environmental Sciences. 4. 41-50.
26. Bowie, J. U., & Sauer, R. T. (1989). Identifying determinants of folding and activity for a protein of unknown structure. *Proceedings of the National Academy of Sciences of the United States of America*, *86*(7), 2152–2156. https://doi.org/10.1073/pnas.86.7.2152
27. Ben-Tal, N., Ben-Shaul, A., Nicholls, A., & Honig, B. (1996). Free-energy determinants of α-helix insertion into lipid bilayers. Biophysical Journal, 70(4), 1803–1812. https://doi.org/10.1016/S0006-3495(96)79744-8
28. Ramachandran, G. N., Ramakrishnan, C., & Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, *7*(1), 95–99. https://doi.org/10.1016/S0022-2836(63)80023-6

29. Harbury, P. B., Tidor, B., & Kim, P. S. (1995). Repacking protein cores with backbone freedom: Structure prediction for coiled coils. *Proceedings of the National Academy of Sciences of the United States of America*, *92*(18), 8408–8412. https://doi.org/10.1073/pnas.92.18.8408
30. Badasyan, A. V., Giacometti, A., Mamasakhlisov, Y. S., Morozov, V. F., & Benight, A. S. (2010). Microscopic formulation of the Zimm-Bragg model for the helix-coil transition. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *81*(2), 1–4. https://doi.org/10.1103/PhysRevE.81.021921
31. Liljas, A., Liljas, L., Piskur, J., Lindblom, G., Nissen, P., & Kjeldgaard, M. *Textbook of Structural Biology*, World Scientific, 2009.
32. Everts, S. (2017). Protein folding, much more intricate than we thought. *American Chemical Society*, Vol. 95, Issue 31, pp. 32-38.
33. Saio, T., Guan, X., Rossi, P., Economou, A., & Kalodimos, C.G. (2014). Structural Basis for Protein Antiaggregation Activity of the Trigger Factor Chaperone. *Science*, *344*(6184), 590–591. https://doi.org/10.1126/science.1250494
34. Sontag, E. M., Vonk, W. I. M., & Frydman, J. (2014). Sorting out the trash: The spatial nature of eukaryotic protein quality control. *Current Opinion in Cell Biology*, *26*(1), 139–146. https://doi.org/10.1016/j.ceb.2013.12.006
35. Verba, K. A., Wang, R. Y. R., Arakawa, A., Liu, Y., Shirouzu, M., Yokoyama, S., & Agard, D. A. (2016). Atomic structure of Hsp90-Cdc37-Cdk4 reveals that Hsp90 traps and stabilizes an unfolded kinase. *Science*, *352*(6293), 1542–1547. https://doi.org/10.1126/science.aaf5023
36. Pechmann, S., & Frydman, J. (2014). Interplay between Chaperones and Protein Disorder Promotes the Evolution of Protein Networks. *PLoS Computational Biology*, *10*(6). https://doi.org/10.1371/journal.pcbi.1003674
37. Balch, W. E., Morimoto, R. I., Dillin, A., & Kelly, J. W. (2008). Adapting proteostasis for disease intervention. *Science*, *319*(5865), 916–919. https://doi.org/10.1126/science.1141448
38. Ovchinnikov, S., Park, H., Varghese, N., Huang, P. S., Pavlopoulos, G. A., Kim, D. E., … Baker, D. (2017). Protein structure determination using metagenome sequence data. *Science*, *355*(6322), 294–298. https://doi.org/10.1126/science.aah4043
39. Ostermann, J., Horwich, A. L., Neupert, W., & Hartl, F. (1989). Protein folding in mitochondria requires complex formation with hsp60 and ATP hydrolysis. *Nature. 341*(September), 125–130.
40. Sarkar, M., Smith, A. E., & Pielak, G. J. (2013). Impact of reconstituted cytosol on protein stability. *Proceedings of the National Academy of Sciences of the United States of America*, *110*(48), 19342–19347. https://doi.org/10.1073/pnas.1312678110
41. Cooper, GM. *The Cell: A Molecular Approach*. 2nd edition. Sunderland, Massachusetts, USA: Sinauer Associates; 2000.
42. Kundra, R., Ciryam, P., Morimoto, R. I., Dobson, C. M., & Vendruscolo, M. (2017). Protein homeostasis of a metastable subproteome associated with Alzheimer's disease. *Proceedings of the National Academy of Sciences of the United States of America*, *114*(28), E5703–E5711. https://doi.org/10.1073/pnas.1618417114
43. Verghese, J., Abrams, J., Wang, Y., & Morano, K. A. (2012). Biology of the Heat Shock Response and Protein Chaperones: Budding Yeast (Saccharomyces cerevisiae) as a Model

System. *Microbiology and Molecular Biology Reviews*, *76*(2), 115–158. https://doi.org/10.1128/mmbr.05018-11
44. Toikkanen, J. (2017). *Expression of proteins in yeast systems*. Retrieved from: http://www.biocenter.helsinki.fi/biotechgs/media/toikkanen.pdf
45. Buckholz, R. G. (1993). Yeast systems for the expression of heterologous gene products. *Current Opinion in Biotechnology*, *4*(5), 538–542. https://doi.org/10.1016/0958-1669(93)90074-7
46. Verma, R., Boleti, E., & George, A. J. T. (1998). Antibody engineering: Comparison of bacterial, yeast, insect and mammalian expression systems. *Journal of Immunological Methods*, *216*(1–2), 165–181. https://doi.org/10.1016/S0022-1759(98)00077-5
47. Baghban, R., Farajnia, S., Rajabibazl, M., Ghasemi, Y., Mafi, A. A., Hoseinpoor, R., … Aria, M. (2019). Yeast Expression Systems: Overview and Recent Advances. *Molecular Biotechnology*, *61*(5), 365–384. https://doi.org/10.1007/s12033-019-00164-8
48. Nielsen, K. H. (2014). Protein expression-yeast. *Methods in Enzymology* (1st ed., Vol. 536). https://doi.org/10.1016/B978-0-12-420070-8.00012-X
49. Van Craenenbroeck, K., Vanhoenacker, P., & Haegeman, G. (2000). Episomal vectors for gene expression in mammalian cells. *European Journal of Biochemistry*, *267*(18), 5665–5678. https://doi.org/10.1046/j.1432-1327.2000.01645.x
50. Lambertz, C., Garvey, M., Klinger, J., Heesel, D., Klose, H., Fischer, R., & Commandeur, U. (2014). Challenges and advances in the heterologous expression of cellulolytic enzymes. *Biotechnology for Biofuels*, *7*(1), 1–15. https://doi.org/10.1186/s13068-014-0135-5
51. Srinivasan, S., Rudolph, D., Durham, D., Heifetz, A. (2006). Expression of Recombinant Proteins in Yeast. *BioPharm International*, *2006*(1). Retrieved from https://www.biopharminternational.com/expression-recombinant-proteins-yeast?id=&pageID=1&sk=&date=
52. Berkner, S., Wlodkowski, A., Albers, S. V., & Lipps, G. (2010). Inducible and constitutive promoters for genetic systems in Sulfolobus acidocaldarius. *Extremophiles*, *14*(3), 249–259. https://doi.org/10.1007/s00792-010-0304-9
53. Lehle, L. (1992). Protein glycosylation in yeast. *Antonie van Leeuwenhoek*, *61*(2), 133–134. https://doi.org/10.1007/BF00580620
54. Oliveira, A. P., & Sauer, U. (2012). The importance of post-translational modifications in regulating Saccharomyces cerevisiae metabolism. *FEMS Yeast Research*, *12*(2), 104–117. https://doi.org/10.1111/j.1567-1364.2011.00765.x
55. Ghaemmaghami, S., Huh, W., Bower, K. et al. Global analysis of protein expression in yeast. Nature 425, 737–741 (2003). https://doi.org/10.1038/nature02046
56. Wang, Q., Xue, H., Li, S., Chen, Y., Tian, X., Xu, X., & Fu, Y. V. (2017). A method for labeling proteins with tags at the native genomic loci in budding yeast. *PLoS ONE*, *12*(5), 1–15. https://doi.org/10.1371/journal.pone.0176184
57. Botstein, D., & Fink, G. R. (2011). Yeast: An experimental organism for 21st century biology. *Genetics*, *189*(3), 695–704. https://doi.org/10.1534/genetics.111.130765
58. Picot J, Guerin CL, Le Van Kim C, Boulanger CM (March 2012). "Flow cytometry: retrospective, fundamentals and recent instrumentation". Cytotechnology. 64 (2): 109–30. doi:10.1007/s10616-011-9415-0

59. Özgür, A., Vu, T., Erkan, G., & Radev, D. R. (2008). Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *Bioinformatics*, *24*(13), 277–285. https://doi.org/10.1093/bioinformatics/btn182
60. Rocklin, G. J., Chidyausiku, T. M., Goreshnik, I., Ford, A., Houliston, S., Lemak, A., … Baker, D. (2017). Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, *357*(6347), 168–175. https://doi.org/10.1126/science.aan0693
61. Li, K., & Malik, J. (2016). *Learning to Optimize*. Retrieved from http://arxiv.org/abs/1606.01885 (About algorithms)
62. Fukuchi, S., Hosoda, K., Homma, K., Gojobori, T., & Nishikawa, K. (2011). Binary classification of protein molecules into intrinsically disordered and ordered segments. *BMC Structural Biology*, *11*(1), 29. https://doi.org/10.1186/1472-6807-11-29
63. Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., & WOO, W. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28* (pp. 802–810). Retrieved from http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting
64. Choi, J. H., Kim, S., Tang, H., Andrews, J., Gilbert, D. G., & Colbourne, J. K. (2008). A machine-learning approach to combined evidence validation of genome assemblies. *Bioinformatics*, *24*(6), 744–750. https://doi.org/10.1093/bioinformatics/btm608
65. Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., … Robles, V. (2006). Machine learning in bioinformatics. *Briefings in Bioinformatics*, *7*(1), 86–112. https://doi.org/10.1093/bib/bbk007
66. Ballester, P. J., & Mitchell, J. B. O. (2010). A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics*, *26*(9), 1169–1175. https://doi.org/10.1093/bioinformatics/btq112 (Protein binding, prediction scores)
67. Williams, N., Zander, S., & Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *Computer Communication Review*, *36*(5), 7–15. https://doi.org/10.1145/1163593.1163596
68. Wang, J., Zhang, Z., & Zha, H. (2005). Adaptive manifold learning. *Advances in Neural Information Processing Systems*.
69. Hamp, T., & Rost, B. (2015). More challenges for machine-learning protein interactions. *Bioinformatics*, *31*(10), 1521–1525. https://doi.org/10.1093/bioinformatics/btu857
70. Kanter, J. M., & Veeramachaneni, K. (2015). Deep feature synthesis: Towards automating data science endeavors. *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015*, 1–10. https://doi.org/10.1109/DSAA.2015.7344858
71. Atzberger, P. J. (2018). *Importance of the Mathematical Foundations of Machine Learning Methods for Scientific and Engineering Applications*. (January). Retrieved from http://arxiv.org/abs/1808.02213
72. Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *ACM International Conference Proceeding Series*, *148*, 161–168. https://doi.org/10.1145/1143844.1143865

73. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236
74. Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., & Kording, K. P. (2019). A deep learning framework for neuroscience. Nature Neuroscience, 22(11), 1761–1770. https://doi.org/10.1038/s41593-019-0520-2
75. Bzdok, D., Altman, N., & Khungar, V. (2018). Statistics versus machine learning. *Nature Methods*, *15*(4), 233–234. https://doi.org/10.1038/nmeth.4642
76. James, G., Witten, D., Hastie T., & Tibshirani R., 2017, *An Introduction to Statistical Learning*, Springer, New York City.
77. Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539
78. Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2017). Deep Learning for Health Informatics. *IEEE Journal of Biomedical and Health Informatics*, *21*(1), 4–21. https://doi.org/10.1109/JBHI.2016.2636665
79. Woolfson, D. N., Bartlett, G. J., Burton, A. J., Heal, J. W., Niitsu, A., Thomson, A. R., & Wood, C. W. (2015). De novo protein design: How do we expand into the universe of possible protein structures? *Current Opinion in Structural Biology*, *33*, 16–26. https://doi.org/10.1016/j.sbi.2015.05.009
80. Liang, P., Yang, W., Chen, X., Long, C., Zheng, L., Li, H., & Zuo, Y. (2020). Machine Learning of Single-Cell Transcriptome Highly Identifies mRNA Signature by Comparing F-Score Selection with DGE Analysis. *Molecular Therapy - Nucleic Acids*, *20*(June), 155–163. https://doi.org/10.1016/j.omtn.2020.02.004
81. Tasic, B., Yao, Z., Graybuck, L. T., Smith, K. A., Nguyen, T. N., Bertagnolli, D., … Zeng, H. (2018). Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, *563*(7729), 72–78. https://doi.org/10.1038/s41586-018-0654-5
82. Fukuchi, S., Hosoda, K., Homma, K., Gojobori, T., & Nishikawa, K. (2011). Binary classification of protein molecules into intrinsically disordered and ordered segments. *BMC Structural Biology*, *11*(1), 29. https://doi.org/10.1186/1472-6807-11-29
83. Younger, D., Berger, S., Baker, D., & Klavins, E. (2017). High-throughput characterization of protein–protein interactions by reprogramming yeast mating. *Proceedings of the National Academy of Sciences of the United States of America*, *114*(46), 12166–12171. https://doi.org/10.1073/pnas.1705867114
84. Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S., & Baker, D. (2020). Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 201914677. https://doi.org/10.1073/pnas.1914677117
85. Jing, X., Dong, Q., HONG, D., & Lu, R. (2019). Amino acid encoding methods for protein sequences: a comprehensive review and assessment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *5963*(c), 1–1. https://doi.org/10.1109/tcbb.2019.2911677
86. Koga, N., Tatsumi-Koga, R., Liu, G., Xiao, R., Acton, T. B., Montelione, G. T., & Baker, D. (2012). Principles for designing ideal protein structures. *Nature*, *491*(7423), 222–227. https://doi.org/10.1038/nature11600

87. Huang, P. S., Ban, Y. E. A., Richter, F., Andre, I., Vernon, R., Schief, W. R., & Baker, D. (2011). Rosettaremodel: A generalized framework for flexible backbone protein design. *PLoS ONE*, *6*(8). https://doi.org/10.1371/journal.pone.0024109
88. Ovchinnikov, S., Park, H., Kim, D. E., DiMaio, F., & Baker, D. (2018). Protein structure prediction using Rosetta in CASP12. *Proteins: Structure, Function and Bioinformatics*, *86*(2003), 113–121. https://doi.org/10.1002/prot.25390
89. Bhardwaj, G., Mulligan, V. K., Bahl, C. D., Gilmore, J. M., Harvey, P. J., Cheneval, O., & Baker, D. (2016). Accurate de novo design of hyperstable constrained peptides. *Nature*, *538*(7625), 329–335. https://doi.org/10.1038/nature19791
90. Leaver-Fay, A., O'Meara, M. J., Tyka, M., Jacak, R., Song, Y., Kellogg, E. H., … Kuhlman, B. (2013). Scientific benchmarks for guiding macromolecular energy function improvement. In *Methods in Enzymology* (1st ed., Vol. 523). https://doi.org/10.1016/B978-0-12-394292-0.00006-0
91. Hoover, D. M. (2002). DNAWorks: an automated method for designing oligonucleotides for PCR-based gene synthesis. *Nucleic Acids Research*, *30*(10), 43e – 43. https://doi.org/10.1093/nar/30.10.e43
92. Cherf, G. M. & Cochran, J. R. (2015). Yeast surface display: Methods, protocols, and applications. *Applications of yeast surface display for protein engineering*, (1319), 155–175. https://doi.org/10.1007/978-1-4939-2748-7
93. Benatuil, L., Perez, J. M., Belk, J., & Hsieh, C. M. (2010). An improved yeast transformation method for the generation of very large human antibody libraries. *Protein Engineering, Design and Selection*, *23*(4), 155–159. https://doi.org/10.1093/protein/gzq002
94. Jacoby, K., Lambert, A. R., & Scharenberg, A. M. (2017). Characterization of homing endonuclease binding and cleavage specificities using yeast surface display SELEX (YSD-SELEX). *Nucleic Acids Research*, *45*(3), e11. https://doi.org/10.1093/nar/gkw864
95. Zhang, J., Kobert, K., Flouri, T., & Stamatakis, A. (2014). PEAR: A fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics*, *30*(5), 614–620. https://doi.org/10.1093/bioinformatics/btt593
96. Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate*. arXiv preprint arXiv:1409.0473.
97. Mueller, J., Gifford, D., & Jaakkola, T. (2017). Sequence to better sequence: Continuous revision of combinatorial structures. *34th International Conference on Machine Learning, ICML 2017*, *5*, 3900–3916.
98. Sinai, S., Kelsic, E., Church, G. M., & Nowak, M. A. (2017). *Variational auto-encoding of protein sequences*. 1–6. Retrieved from http://arxiv.org/abs/1712.03346
99. Bileschi, M. L., Belanger, D., Bryant, D., Sanderson, T., Carter, B., Sculley, D., & Colwell, L. J. (2019). Using Deep Learning to Annotate the Protein Universe. *BioRxiv*, 1–29. https://doi.org/10.1101/626507