Developing Computational Tools for Improving Protein Stability and Increasing Protein Binding Affinity

by

Alexey Strokach

Contents

1 Overview **1**

1.1 Discussion 2

1.1.1 Unresolved questions 2

1.2 Future directions 2

Bibliography **4**

List of Tables

List of Figures

1.1 An exemplary image crazy crazy text. Hello world 4
1.2 Recurrent Geometric Network. 5

Overview

Traditional methods of predicting residue residue contacts. These advances have largely been driven by the use of residue coevolution, deduced from multiple sequence alignments, in order to predict interactions. However, their reliance on evolutionary information precludes the application of these approaches to the design of entirely new proteins, with no existing homologs found in nature.

There is strong interest in new tools and approaches for improving protein stability and increasing the affinity of proteins to their targets. In academic research laboratories, such tools could be used to design proteins that are easier to crystallize and that show higher yield and are less prone to aggregation when expressed in heterologous systems ([1, 2]). In industrial process engineering, such tools could be used to design biocatalysts which remain active in inhospitable conditions, including high temperatures and pressures, acidic or basic pH, different salt types and concentrations, and admixture of organic solvents ([3]). In the pharmaceutical industry, such tools could be used to optimize protein and peptide "biologics" to improve their shelf life and increase their affinity and specificity to desired targets ([4]).

Existing approaches for protein optimization and design are either too computationally expensive, too inaccurate, or both, to be practical for routine use. Alchemical free energy calculations based on molecular dynamics can accurately predict the change in the Gibbs free energy of folding or binding ($\Delta\Delta G$) associated with mutations [5, 6]. However, evaluating the impact of a single mutation can take hours to days on a modern graphical processing unit, making this approach intractable for protein design. Statistical potentials can be used to evaluate a large number of amino acid substitutions proposed by Markov chain Monte Carlo or other sampling techniques [7, 8]. However, statistical potentials are not very accurate, and the sampling procedure generally keeps the backbone of the protein fixed and thereby only selects sequences around a local energy minimum of a single protein conformation. To date, most successful examples of computational protein design have involved a final step where hundreds of thousands of designed sequences are screened using high-throughput experimental techniques to select a handful of sequences that produce proteins with notably improved properties [9, 10].

The overarching goal of my PhD project is to develop computational methods that can be used to construct libraries of proteins that have higher stability, higher catalytic activity, or higher affinity to their target than the starting structure. Two recent developments can help me in achieving this goal. First, advances in next-generation sequencing technology have led to enormous growth in the amount of genomic data that is available, and this data has already been shown to be useful in structural biology. For example, in our work on ELASPIC [11], we observed that using a mutation deleteriousness score in addition to structural information can improve the accuracy with which we can predict the energetic effect of mutations. As another example, co-evolutionary information extracted from multiple sequence alignments has been used to accurately predict the three-dimensional structure of proteins [12,

13], suggesting that this data can also be immensely useful for protein design. Second, advances in neural networks, coupled with the growth in the volume of data and the computational resources that are available, have led to striking breakthroughs in many fields such as image detection and natural language processing. While there has been some work on using neural networks for protein optimization and design [14, 15], this application remains relatively under-explored and much headway remains to be made.

The report is structured as follows. In Chapter **??**, we discuss how molecular dynamics simulations can be used to evaluate the energetic impact of mutations on protein folding and binding. In Chapter **??**, we discuss the development and optimization of statistical potentials and the application of statistical potentials to protein design. In Chapter **??**, we examine a study which uses recurrent neural networks, trained on a dataset of sequential and structural information, to "fold" proteins into their three-dimensional shape. In Chapter **??**, we examine a study which uses generative adversarial neural networks to optimize existing DNA sequences and generate new DNA sequences with desirable properties. Finally, in Chapter **??**, we outline our progress in training a neural network to predict whether a given sequence matches a given adjacency matrix. We also discuss the difficulties that we are facing in training a generative adversarial network to generate sequences that fold into a given shape.

1.1 Discussion

For my PhD project, I propose to examine how recent advances in deep learning can be used to improve the problem of protein design.

There are two hurdles that limit our ability to stabilize proteins: insufficient accuracy of the energy functions that are used to score different conformations, and inability to model the motion of proteins quickly.

There is still a limit of how accurate of results we can achieve with a fixed backbone method. We could explore several approaches for introducing backbone flexibility:

```
A) Select several different PDB templates from which to extract adjacency matrices. B) Include co-e
```

1.1.1 Unresolved questions

1.2 Future directions

1.2

Our immediate goal is to evaluate further the accuracy of the classifier network described in Section **??** and to publish this network, which we call the Protein Adjacency Graph Neural Network or PAGNN, in a standalone article. We have already shown that PAGNN can accurately predict whether a given sequence matches a given adjacency matrix for a set of Gene3D domains that were not present in the training dataset (Figure dcn*training*curvesB), and we have shown that there is a strong correlation between the difference in PAGNN predictions for the wild-type and mutant sequences and the $\Delta\Delta G$ associated with the mutations (Figure dcn*training*curvesC). We are in the process of evaluating the utility of PAGNN on several additional tasks, including decoy discrimination, remote homology detection, and homology modeling. In the decoy discrimination task, we will assess the ability of PAGNN to pick out a real structure from a set of decoys generated using 3DRobot [16] or a similar method. We will also

correlate the PAGNN score with the root-mean-square distance of the decoy structures to the reference structure. In the remote homology detection task, we will assess the ability of PAGNN to improve the assignment of a given sequence to the correct SCOPe family. First, we will use `hhpred` to construct a set of alignments of the query sequence to its closest homologs in the SCOPe database. Next, we will use PAGNN to score each of the alignments using the adjacency matrix of the homologous protein, and we will assess the ability of PAGNN to rank more highly the alignments to proteins that belong to the correct SCOPe family. In the homology detection task, we will assess the ability of PAGNN to improve the quality of the homology models that are constructed for a given sequence. The primary factor affecting the quality of a homology model is the accuracy of the alignment of the query sequence to the structural template [17]. We will use PAGNN to score each of the alignments produced by `hhpred` and we will correlate the scores that PAGNN assigns to those alignments with the quality of the resulting homology models.

[**hellwoorld**].

Our subsequent goal will be to optimize the generative adversarial network (GAN) described in Section **??** to achieve sufficient accuracy to be used for constructing libraries that can be screened using high-throughput experimental techniques [9, 10]. Training the GAN using real and permuted sequences (Table internal*validation*datasets) results in a generator that can produce sequences with roughly the correct secondary structure but incorrect orientation of the secondary structure elements with respect to one another (see Figures gan*validation*seq, gan*validation*ss, and gan*validation*models). Therefore, we have to modify our training dataset to make the classification task more difficult. We also need to address some of the issues with the architecture of our network. While using batch normalization is reported to significantly speed up and improve the stability of the training process, in our case it results in an unwieldy network that is difficult to validate and to use after training. We can instead try using instance normalization [18], layer normalization [19], or weight normalization [20], which do not have the same shortcomings. Once we have a GAN that can reliably generate sequences, we can use additional filters to select a subset of the sequences that are the most likely to fold into a desirable shape. One approach would be to create homology models of each of the sequences and to sort those homology models using DOPE scope, TM-Score, or the Rosetta scoring function. - It remains unclear whether including structural information leads to better accuracy than training a transformer model on sequence input only [20] [20].

1.2

Text [1]. [2]

```
import os

print("hello world")
```

Reference Figure 1.1.

> Quoting from some famous person.

A link to a website https://foo.bar.

This is **important**.

[1] This is a footnote.

[2] some crazy footnote somethng

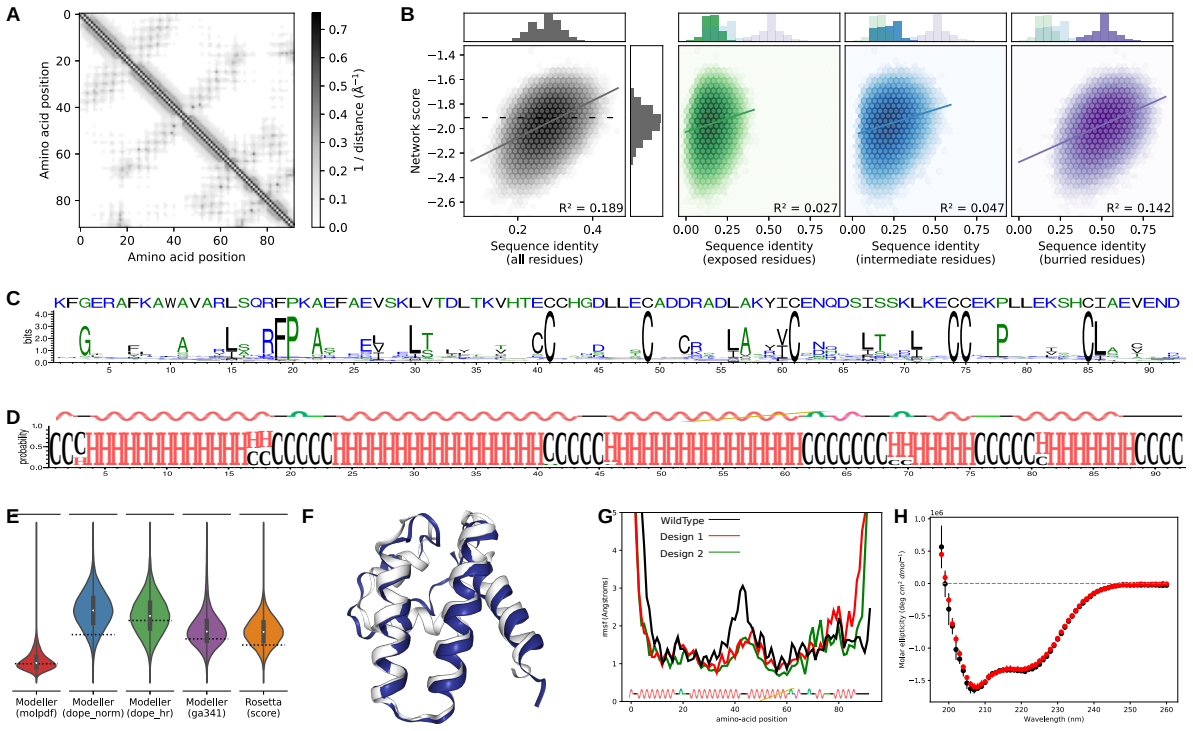

Figure 1.1: An exemplary image crazy crazy text. Hello world

Not as important as ***this***. Don't forget *this*.

$$E = mc^2 \tag{1.1}$$

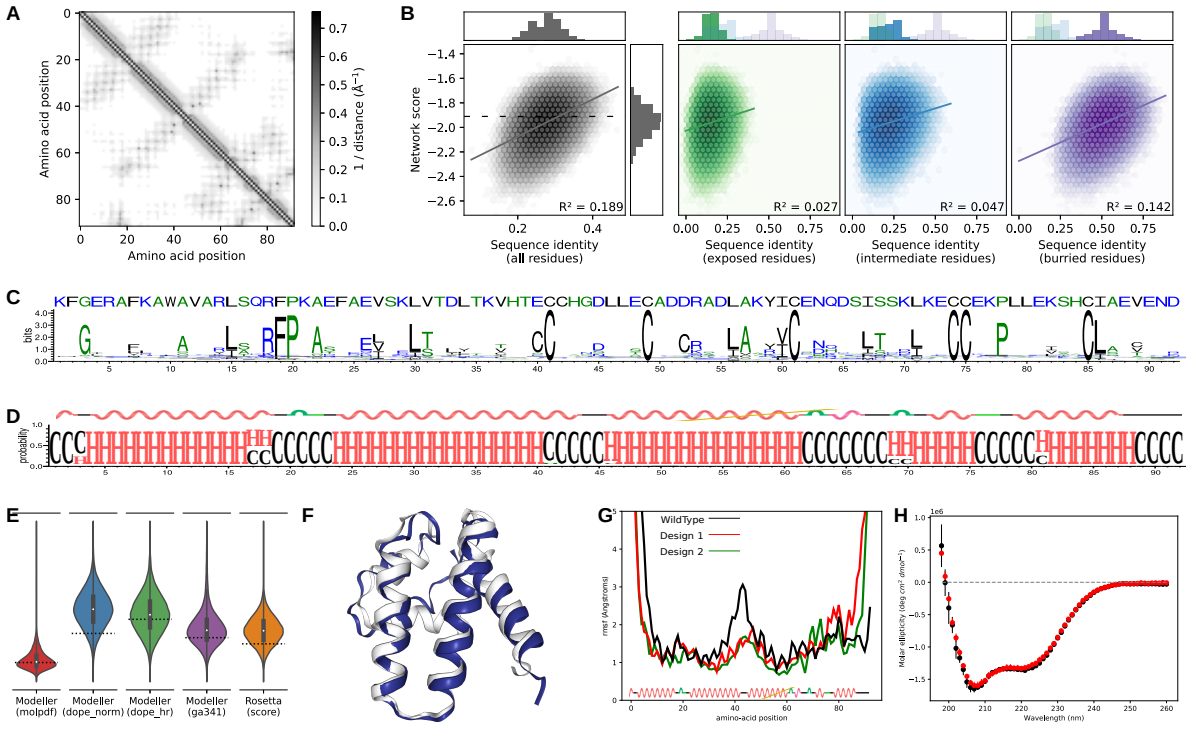

Figure 1.2: Diagram of a Recurrent Geometric Network (RGN) (adapted from hello world AlQuraishi [15]).

Bibliography

[1] M. C. Deller et al. "Protein Stability: A Crystallographer's Perspective". In: *Acta Crystallographica Section F: Structural Biology Communications* 72.2 (February 1, 2016), pp. 72–95. DOI: 10.1107/S2053230X15024619.

[2] Petr Popov et al. "Computational Design of Thermostabilizing Point Mutations for G Protein-Coupled Receptors". In: *eLife* 7 (June 21, 2018), e34729. DOI: 10.7554/eLife.34729.

[3] Andreas S. Bommarius and Mariétou F. Paye. "Stabilizing Biocatalysts". In: *Chemical Society Reviews* 42.15 (July 8, 2013), pp. 6534–6565. DOI: 10.1039/C3CS60137D.

[4] Marco L. Davila and Renier J. Brentjens. "CD19-Targeted CAR T Cells as Novel Cancer Immunotherapy for Relapsed or Refractory B-Cell Acute Lymphoblastic Leukemia". In: *Clinical advances in hematology & oncology : H&O* 14.10 (October 2016), pp. 802–808.

[5] Daniel Seeliger et al. "Geometry-Based Sampling of Conformational Transitions in Proteins". In: *Structure* 15.11 (November 13, 2007), pp. 1482–1492. DOI: 10.1016/j.str.2007.09.017.

[6] Vytautas Gapsys et al. "Accurate and Rigorous Prediction of the Changes in Protein Free Energies in a Large-Scale Mutation Scan". In: *Angewandte Chemie International Edition* 55.26 (June 20, 2016), pp. 7364–7368. DOI: 10.1002/anie.201510054.

[7] Brian Kuhlman et al. "Design of a Novel Globular Protein Fold with Atomic-Level Accuracy". In: *Science* 302.5649 (November 21, 2003), pp. 1364–1368. DOI: 10.1126/science.1089427.

[8] Joost Schymkowitz et al. "The FoldX Web Server: An Online Force Field". In: *Nucleic Acids Research* 33 (suppl 2 January 7, 2005), W382–W388. DOI: 10.1093/nar/gki387.

[9] Mark G. F. Sun et al. "Protein Engineering by Highly Parallel Screening of Computationally Designed Variants". In: *Science Advances* 2.7 (July 1, 2016), e1600692. DOI: 10.1126/sciadv.1600692.

[10] Gabriel J. Rocklin et al. "Global Analysis of Protein Folding Using Massively Parallel Design, Synthesis, and Testing". In: *Science* 357.6347 (July 14, 2017), pp. 168–175. DOI: 10.1126/science.aan0693.

[11] *Daniel K. Witvliet et al. "ELASPIC Web-Server: Proteome-Wide Structure-Based Prediction of Mutation Effects on Protein Stability and Binding Affinity". In: *Bioinformatics* 32.10 (May 15, 2016), pp. 1589–1591. DOI: 10.1093/bioinformatics/btw031.

[12] Thomas A Hopf et al. "Three-Dimensional Structures of Membrane Proteins from Genomic Sequencing." In: *Cell* 149.7 (2012), pp. 1607–21.

[13] Sergey Ovchinnikov et al. "Protein Structure Determination Using Metagenome Sequence Data". In: *Science* 355.6322 (January 20, 2017), pp. 294–298. DOI: 10.1126/science.aah4043.

[14] Jingxue Wang et al. "Computational Protein Design with Deep Learning Neural Networks". In: *Scientific Reports* 8.1 (April 20, 2018), p. 6349. DOI: 10.1038/s41598-018-24760-x.

[15] Mohammed AlQuraishi. "End-to-End Differentiable Learning of Protein Structure". In: *bioRxiv* (February 14, 2018), p. 265231. DOI: 10.1101/265231.

[16] Haiyou Deng et al. "3DRobot: Automated Generation of Diverse and Well-Packed Protein Structure Decoys". In: *Bioinformatics* 32.3 (February 1, 2016), pp. 378–387. DOI: 10.1093/bioinformatics/btv601.

[17] Marc A. Marti-Renom et al. "Alignment of Protein Sequences by Their Profiles". In: *Protein Science : A Publication of the Protein Society* 13.4 (April 2004), pp. 1071–1087. DOI: 10.1110/ps.03379804.

[18] Xun Huang and Serge Belongie. "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization". In: (March 20, 2017).

[19] Jimmy Lei Ba et al. "Layer Normalization". In: (July 21, 2016).

[20] Tim Salimans and Diederik P. Kingma. "Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks". In: (February 25, 2016).