A Dance with Protein Assemblies

Analysis, Structure Prediction, and Design

Jeppesen, Mads

2023

A Dance with Protein Assemblies

Analysis, Structure Prediction, and Design

Mads Jeppesen

DOCTORAL DISSERTATION

Doctoral dissertation for the degree of Doctor of Philosophy (PhD)
at the Faculty of Science at Lund University to be publicly defended
on Friday, 24th of November at 13.00 in Lecture Hall B,
Department of Chemistry, Naturvetarvägen 14, Lund

Faculty opponent
Björn Wallner

Organization: Lund University

Document name: Doctoral Dissertation **Date of issue:** 2023-11-24

Author(s): Mads Jeppesen **Sponsoring organization:**

Title and subtitle: A Dance with Protein Assemblies: Analysis, Structure Prediction, and Design

Abstract:

Protein assemblies are some of the most complex molecular machines in nature. They facilitate many cellular functions, from DNA replication to molecular motion, energy production, and even the production of other proteins. In a series of 3 papers, we analyzed the structure, developed structure prediction tools, and design tools, for different protein assemblies. Many of the studies were centered around viral protein capsids. Viral capsids are protein coats found inside viruses that contain and protect the viral genome.

In one paper, we studied the interfaces of these capids and their energy landscapes. We found that they differ from regular homomers in terms of the amino acid composition and size, but not in the quality of interactions. This contradicts existing experimental and theoretical studies that suggest that the interactions are weak. We hypothesise that the occlusion by our models of electrostatic and entropic contributions might be at play.

In another paper, we developed methods to predict large cubic symmetrical protein assemblies, such as viral capsids, from sequence. This method is based upon AlphaFold, a new AI tool that has revolutionized protein structure prediction. We found that we can predict up to 50% of the structures of these assemblies. The method can quickly elucidate the structure of many relevant proteins for humans, and for understanding structures relevant to disease, such as the structures of viral capsids.

In the final paper, we developed tools to design capsid-like proteins called cages – structures that can be used for drug delivery and vaccine design. A fundamental problem in designing cage structures is achieving different architectures and low porosity, goals that are important for vaccine design and the delivery of small drug molecules. By explicitly modelling the shapes of the subunits in the cage and matching the shapes with proteins from structural databases, we find that we can create structures with many different sizes, shapes, and porosities - including low porosities. While waiting for experimental validation, the design strategy described in the paper must be extended, and more designs must be tested.

Key words: Protein design, virus capsids, protein structure prediction, symmetry, Rosetta, protein assembly.

Classification system and/or index terms: **Supplementary bibliographical information:**

Language: English **ISSN and key title**:

ISBN: 978-91-7422-990-5 (print)

978-91-7422-991-2 (digital)

Recipient's notes: **Number of pages**:

Price: **Security classification:**

A Dance with Protein Assemblies

Analysis, Structure Prediction, and Design

Mads Jeppesen

LUND
UNIVERSITY

Coverphoto: Protein assemblies falling from the sky. Created by Gencraft.

Faculty of Science

Department of Chemistry

ISBN : 978-91-7422-990-5 (print),

ISBN: 978-91-7422-991-2 (digital)

Printed in Sweden by Media-Tryck, Lund University

Lund 2022

NORDIC SWAN ECOLABEL
Printed matter
3041 0903

Media-Tryck is a Nordic Swan Ecolabel certified provider of printed material. Read more about our environmental work at www.mediatryck.lu.se

MADE IN SWEDEN

To all who’ve guided my journey

Table of Contents

List of publications

The thesis is based on the following papers:

Paper I:

Accurate prediction of protein assembly structure by combining AlphaFold and symmetrical docking

Mads Jeppesen, Ingemar André

In review, Nature communications

Paper II:

Encoding of T=1 virus capsid structures through the interfaces of oligomer subcomponents

Mads Jeppesen, Ingemar André

Manuscript

Paper III:

A method for designing protein cages based on shape.

Mads Jeppesen, Ingemar André

Manuscript

Author Contributions

The authors' contribution to the papers.

Paper I:

The conceptualization and methodology of this study was done in collaboration with I.A. I wrote all software extensions to EvoDOCK, ran all the simulations, and did the formal analysis. The writing of the original draft and editing was done in collaboration with I.A.

Paper II:

The conceptualization and methodology of this study was done in collaboration with I.A. I collected all the data, did parts of the initial analysis, and edited the manuscript in collaboration with I.A.

Paper III:

The conceptualization and methodology of this study was done in collaboration with I.A. The code was written in collaboration with I.A. I ran the simulations to design the protein structures. Evaluation of the structures was done in collaboration with I.A. I wrote the original draft and editing in collaboration with I.A.

Popular summary

Proteins are tiny molecules that cannot be seen with the naked eye. They might appear insignificant, but they have big roles to play in biology and are often dubbed the ‘molecule of life’. They are everywhere in nature: inside of you, inside bacteria, plants, fungi, and viruses. They are nature's answer to molecular machines as they can change shape, catalyze, signal, transport, move, and produce. They are truly remarkable molecules.

As us humans, proteins can be either introverted and like to work alone, or extroverted, and like to associate with other proteins into what we call protein assemblies. This thesis is titled: “A Dance with Protein Assemblies: Analysis, Structure Prediction, and Design”. This is because we analysed, and developed structure prediction tools, and design tools, for different kinds of protein assemblies. We did this in a series of papers, as referenced on the previous pages.

In **Paper II** we danced with a kind of protein assembly found inside viruses called a capsid. A capsid is a protein container that protects the genetic material of the virus, which it uses to replicate inside its host (which could be you!). We wanted to understand how they self-assemble from single proteins into a protein assembly in one of the most complex self-assembly processes in nature (no small feat). We did this by comparing its subcomponents to regular smaller assemblies found elsewhere in nature. While we saw some differences, our main hypothesis was that the strength of the capsid interfaces must have been weaker since capsids have to dissociate and associate when they deliver and repackage their genetic material. However, our data does not support this, and we hypothesize that this ability might come from other contributions not explicitly modelled, such as entropy or electrostatics.

In **Paper I** we took up another dance, riding the wave of a new generation of AI tools that revolutionized biology. In the middle of my PhD, an AI software called AlphaFold, developed by Google's DeepMind group, made a breakthrough in our ability to predict protein structures with high resolution. This is important, as understanding the structure of proteins is the key to understanding their function, implications in diseases, and how we can develop drugs against them. However, AlphaFold struggled to predict the structure of protein assemblies, in particular ones with many chains (the very extroverted ones). After years of dancing with protein assemblies, we had developed tools for which we could build upon AlphaFold to predict them with high resolution. We did just that and showed we could predict many important classes of large protein assemblies, including virus capsids described previously.

In the longest and still ongoing dance (**Paper III**), we developed a design software to design protein cages, a container-like virus capsid, that can be used to encapsulate drugs for targeted delivery in the body or for highly efficient vaccines. It’s built on a novel principle we call shape-based design. The idea is first to identify shapes that fit into such a container, followed by alignment of proteins with those exact shapes into the container. We designed different architectures, which subsequently were validated to work well computationally and have characteristics like natural proteins. However, experimental testing is still in an ongoing process, and we haven’t had the last dance yet.

Acknowledgements

Embarking on a PhD is no small endeavour. While you attempt to walk where no other human has walked before, you must deal with failure, disappointment, and feelings of inadequacy. While nothing works and you feel like you are heading nowhere, you still have to show up, put your best foot forward, and try to figure it out. When you are knee-deep in the scientific trenches, it is paramount to have people around you who can support you, encourage you, and give you a sense of relief from the everyday scientific battle. This thesis is dedicated to all of you who have helped me on the journey. Not only in science but in life at large. To former and current teachers, colleagues, friends, and family.

First and foremost, I need to thank my supervisor **Ingemar**. Thank you for showing interest in me even way before I started in your group. For helping me out with my research project in another group, in another country, giving me career advice and allowing me to come and visit back in 2018 when I was still a Master student. You are genuinely interested in helping other budding scientists and expects nothing in return. Thank you for the last five years, guiding my way through the dark corners of science, showing patience, interest, kindness, and a few good jokes along the way. Thank you for believing in me and that I was someone cut out for doing a PhD and thank you for entrusting your research project to me. I still hope to achieve what we set out to do all these years ago!

To former members of the André lab. **Christoffer**, thank you for being an all-around great dude and a scientific role model who helped me tremendously before and in the humble beginnings of my PhD. To **Filip**, even though you are a MATLAB user you are still a good dude with a beautiful beard. Thank you for the shape talks and the contribution you gave to the shape project. **Signe**, my Danish compatriot, thank you for as well for your contribution to the shape project, helping build a great atmosphere in the lab, and make me feel less alone behind enemy lines. **Morteza**, I still dream of the Iranian food you and your family cooked up for us some years ago – also thank you for saving my life alongside Filip on the dreaded waters outside Ven. **Teun**, you were a great person to have in the lab - thank you for the good vibes, I really miss you. You were asking some tough Rosetta questions, that had me dive into places in the codebase no one has ever been before. To **Daniel (**or David?), I miss your companionship and someone who also swears and shouts when you write code. I feel weird and alone now. Also, thank you for creating the docking software to rule all docking software. I hope other people in the docking community will see the light someday. To **Caroline**, **Iva**, **Sofie** and **Valdemar,** who are also integral to the shape project, I thank you for investing your time and for the good times we had. To **Ryan**, **Fabio**, **Tao**, **Arthi** thank you as well for the good times – and thank you Fabio for being a great swimming companion. And who can forget **Ted** (or Teb?) **-** the best roommate a traveller can ever ask for and a solid dude and politician. Please remember me when Volt takes over the EU. And to **Lucas** - great hat, almost as great as you!

To current members of the lab. **Wojtek**, thank you for being a genuinely good person who really cares about the wellbeing of other people in the lab. I never forget the trip you took me on to Skrylle to hang out and eat cake. It was needed in hard times. **Camille,** it has been such a pleasure to share the PhD journey with you. Thank you for always being kind, funny and a great person to talk to. Thank you for feeding me with various things. I genuinely wish you success moving forward. You are a great scientist and I know you'll be successful! Also thank you for your contribution to the shape project. To **Nillergøj,** you have made my PhD so much more fun, and I can never thank you enough for all the laughs you have brought me. You have an impeccable taste for memes, you are a great friend and thank you for accepting my weirdness. To **Sofia**, thank you for reviving the coffee machine, and making sure it always smells lovely in the morning. Looking forward to more vacation/conference presentations! **Vera**, thank you for the great humour and breathing life into EvoDOCK!

To the newer members of André lab. **Helene**, I can tell I'm already going to learn a lot of deep learning from you! **Aswathy,** remind me to repeat a joke Niels said was excellent**.** **Felipe,** thanks for showing a lot of trust in me for potentially lighting your whole mouth on fire**.** Also, thanks for calling me a Giga Chad. Now I believe.

To people outside André lab. **Daniel**. Tack so mycke for at have lært mig det svenska språk. Utan dig vil jeg aldrig have lært det. Och tack så mycke for at invitere mig til midsommer, det var så snyggt og roligt! Jeg skal snart tage dig I gym og vise dig hvordan man træner dine kyllinge-arme. To **Carmy**, my climbing partner and now bike partner. Let's keep making the roads and walls of Lund and Malmø unsafe (I'm trying at least). I'm so happy you decided to stay for a PhD - you make the department a so much better place for everyone! To **Constantinos**. If I had half your game, I would be swimming in the ladies by now. Let's go to a water park soon. To **Simon,** I never had so much fun dancing with another person. You are awesome. I miss the movie days. Speaking of movie days. To **Helin,** thank you for being such a fun and empathic person. Please stay for at least one beer club in the future! You make the department a great place to work, with perhaps a reduction in productivity… To **Emil**, you are a very inspirational and adventurous guy. I really hope we can climb some mountains and cross some rivers together in the future. Also, we should go on roller skis soon. To **Magnus**, you keep the department safe and sound and are always good for a laugh and a joke! To **Maryam**, you keep the spirit of the department up. It's been great to have you around! To **Max**. I love your passion for beer and keeping BC alive. You are very needed in the department. Also, when Niels gets better at Karate you two should fight. I wager 3 Tuborg on Niels. To **Lei,** thanks for keeping KEMCO3 alive with me and for reminding me there is no bad weather, only shorts. To **Johan**, one of the only Swedes I entrust speaking Danish to. Thank you for all the running advice and the good conversations. Let's go for a run after this. To **Jing.** Still sorry about the hike. You are a fun person and I love to climb with you! To **Kristine**, det er så godt at have en anden dansker at holde fanen op med, og snak li't cykling - så'en nået de skøre svensker ikke ve' nå't om. To **Alex**, thank you for being a great roomy. To **Nicholas** for great cycling advice. To **Eimantas –** I don't know why we never talk more about Århus**?** **Lovisa**. Remind me to give me some hiking advice. **Andreas,** hope your tennis arm is doing better. Still disappointed you muted the climbing chat… To **Egle,** thanks for the convos and for making me feel a little more at ease with the thesis writing. To the always kind **Balder**, look forward to hosting Dora with you. To **Angus,** 1 letter away from being a legend**.** Fortunately, you already are one. **Erik,** you are super cool 10/10. **Dev –** cool tattoo**!** To **Karoline, Lorenza, Simon, Victoria, Mattias** and to all the people I have yet to meet – I'm looking forward to getting to know you better! (I'm sticking around for a while suckers)

To all the professors, researchers and teaching staff: **Akke**, **Sara**, **Tommy**, **Henrik** (please start in comedy when you retire from science), **Urban**, **Susanna**, **Thom** (I want to be an

ultra-marathoner like you when I grow up), **Derek**, **Modig**, **Herwig (**Hope you got the server up working**)**, **Pär** (Thank you for the discussions on thermodynamics and co-supervising) and the KEMCO3 teammates: **Martin** and **Katja!** Thank you for a harbouring a great department and entrusting us PhD-students and the like to help with your science.

To people who have left the department for new horizons. To **Olof** and **Christin** – thank you for all the runs. **Mathias** for the best Christmas party teammate! It will never be repeated. Also, the entire KEMCO3 course hanged on your shoulders. You made me a computer scientist feel at ease in the lab. To **Bhakat**, miss you, thank you for opening to me and being an honest good dude. To **Dr. Butt** for the beers. Thank you for entrusting me with the BC club. I'm sure it wasn't for my alcoholism but for my professionalism. To **Gemma**. I'm so happy to climb with you and look forward to seeing your progress! To **Antonio**. Parties have never been the same since you left - see you for the next beer. To **Yonathan**. Thank you for enriching my lacklustre movie-culture taste. To **Sven**, I thank you (and Camille) for your friendship. I miss the old days. To **Vero,** thank you for your emotional support**,** being truly yourself, and your friendship. To **Celia**, your German BRAUSE BONBONS is by my side as I write. I'm on my 20th piece of the night and I deeply regret my decision. I miss you and sincerely hope you'll have a great time in Denmark. To **Mathieu**, I miss your laugh and your love for Disney. **Archimede** your Italian vibes are missed. To **Virginia**, my Spanish climbing partner, hiking partner, running partner and neighbour/stalker? I had such a great time when you were here. You have a great energy and outlook on life! I hope Dresden is treating you well. I miss you. To **CJ (**and **Hafsa)**. If a Dane could ever call a Swede a friend (unimaginable I know), you would be it. You made my time in Sweden infinitely better and I look forward to all the future adventures we can have together. See you in Stockholm!

There are so many more people who deserves a shoutout, **Rebekka**, **Jelica**, **Rohit**, **Tinna**, **Yulian,** and many others – Sorry if I forgot you, thesis writing messes up your brain.

To a person we unfortunately lost. To **Ipsita**. You were a wonderful person, and you are truly missed.

To people who have never stepped a foot in the department. To the **Rosetta community** for being a great group of excited and inclusive scientists. To the countless **physiotherapist** who has duck taped my body together. This thesis would not have been physically possible without you.

To my **friends** back in Denmark. Thank you for being there for me and letting me into your lives and accepting me for who I am. Thank you to my **family** for raising me and supporting me always. I don't think I can ever thank you enough.

To 1STM. You were the best model system a protein designer could ever wish for. I love you.

Abbreviations

AF	AlphaFold
AF2	AlphaFold2
AFM	AlphaFold-Multimer
AI	Artificial intelligence
CAPRI	Critical Assessment of Predicted Interactions
CASP	Critical Assessment of Structure Prediction
CPD	Computational protein design
Cryo-EM	Cryo-electron microscopy
EA	Evolutionary algorithm
FFT	Fast Fourier Transform
FM	Free Modelling
DNA	DeoxyriboNucleic Acid
DE	Differential Evolution
DL	Deep Learning
MC	Monte Carlo
MCTS	Monte Carlo Tree Search
MD	Molecular Dynamics
MM	Molecular Mechanics
mRNA	messenger RNA
MSA	Multiple Sequence Alignment
NLP	Natural Language Processing
NMA	Normal Mode Analysis
NMR	Nuclear Magnetic Resonance
NN	Neural Network
NSR	Native Sequence Recovery
PDB	Protein Data Bank
PPI	Protein-Protein Interface
RNA	RiboNucleic Acid
SASA	Solvent accessible surface area
SC	Shape Complementarity
TBM	Template-Based Modelling
T-number	Triangulation number
ZC	Zernike-Canterakis
ZCD	Zernike-Canterakis Descriptors

Introduction

Protein assemblies are some of the most complex molecular machines in nature. They facilitate many cellular functions from DNA replication to molecular motion, to ATP production, and even the production of other proteins. This thesis is titled: “A Dance with Protein Assemblies: Analysis, Structure Prediction, and Design”. That is because we in a series of different papers analyzed (**Paper II**), and developed structure prediction tools (**Paper I**) and design tools (**Paper III**) for different types of protein assemblies. Although different, they do however share a deeper relationship as you’ll discover. All the papers can be found in the back in order from **I** to **III.**

Before we arrive at the three papers, we need to dive into the background and put them into context. There are 8 chapters in this thesis:

- Chapter 1: Protein Structure
- Chapter 2: Computational Modelling of Proteins (☕, ☕, ☕)
- Chapter 3: Protein Structure Prediction (☕)
- Chapter 4: Protein Design (☕)
- Chapter 5: Summary of Research Papers
- Chapter 6: Additional Background (☕, ☕)
- Chapter 7: Conclusion
- Chapter 8: Outlook

In Chapter 1, I will discuss the general structure of proteins and the structure of virus capsids. This will lay the foundation for the rest of the thesis and give background to **Paper II**. In Chapter 2, I will discuss computational methods relevant to protein structure prediction and design. Similarly, to Chapter 1, it will serve as a background for the rest of the thesis. Chapters 3 and 4 are more specific chapters targeted at the general topics and literature for each paper. Chapter 3 relates to **Paper I** and Chapter 4 to **Paper III**. Chapter 5 summarizes the findings and future work of each paper. Chapter 6 serves as additional background information for **Paper I** and **III**. The final chapters, Chapters 7 and 8 will conclude the thesis and give an outlook on the fields and themes we have discussed.

As you might have noticed the chapters are marked with coffees (☕). The numbers of ☕ indicate how much coffee is needed to be consumed to get through them. Don’t underestimate them, these are like the chilis you find on the menu list in a good authentic Chinese restaurant.

Enjoy!

1 Protein Structure

In this chapter, we will examine the structure of proteins from their constituents, the amino acids, to protein folding, structural classification, and symmetry. It will lay the foundation for the chapters to come. The chapter will end with a discussion on viruses, and one of their components called a capsid: a protein assembly encapsulating and protecting the viral genome.

1.1 Amino acids

Proteins are polymers that consist of amino acids linked together in a sequence like beads on a string. Each amino acid consists of an invariant part called the backbone and a variant part called the side chain. The backbone contains an amine and carboxyl group covalently linked to a central carbon atom. As the backbone is the same for all amino acids, the side chain makes them unique, and their chemical groups determine their shape and chemistry. **Figure 1a** shows all the 20 different amino acids, divided into chemical categories depending on their side chains. To create a protein, amino acids are linked through their backbone by a condensation reaction to form peptide bonds, linking one amino acid carboxyl group to another's amine group with water as a by-product (**Fig 1b**). Amino acids can be synthesized by an organism itself or acquired by breaking down proteins from other organisms. For instance, eleven amino acids can be synthesized in humans, but nine can not[1]. The amino acid sequence of a protein is encoded by genes: three base pair segments in DNA called a codon, encode for either one amino acid or for a start or stop signal. To synthesize a protein, a gene is first transcribed into mRNA and protein factories called ribosomes then translate this mRNA code to produce a full protein.

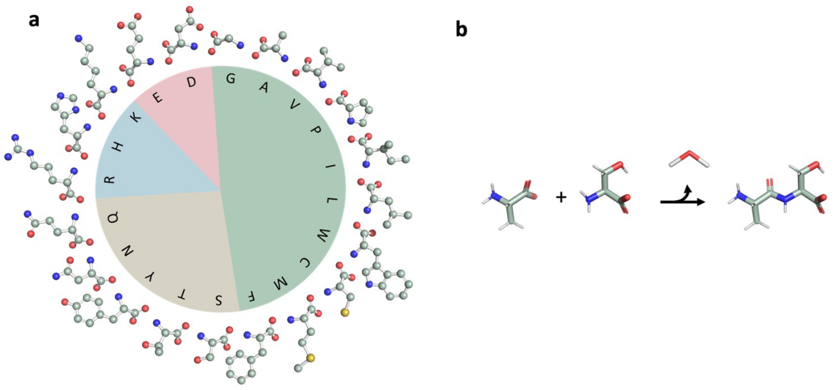

Figure 1: The 20 canonical amino acids and peptide bonds through condensation.
a: The chemical structure and one-letter name of the 20 canonical amino acids that are encoded by codons. Each amino acid has a different side chain and is grouped into four groups based on their chemistry as highlighted by their adjacency to a respective color. In green, we have the hydrophobic side chains, and in brown the hydrophilic side chains. In blue we have the positively charged side chains and in red the negatively charged side chains. **b**: Condensation reaction covalently bonding amino acid A (alanine) to the amino acid S (serine) through a peptide bond.

1.2 Protein Folding

Once a protein has been translated, it needs to fold up to a specific conformation, called the native structure to carry out its function. However, even for a very simple protein, there is an astronomical number of possible conformations available. How can a protein find the native state out of all the available states? According to the laws of thermodynamics, the protein will approach the state with minimal Gibbs free energy (G) which is described by:

$$G = H - TS \tag{1.1}$$

H is the enthalpy (internal energy), S is the entropy, and T the temperature. To see how a protein can minimize its free energy it is helpful to visualize a simplified free energy landscape as in the example shown in **Figure 2.**

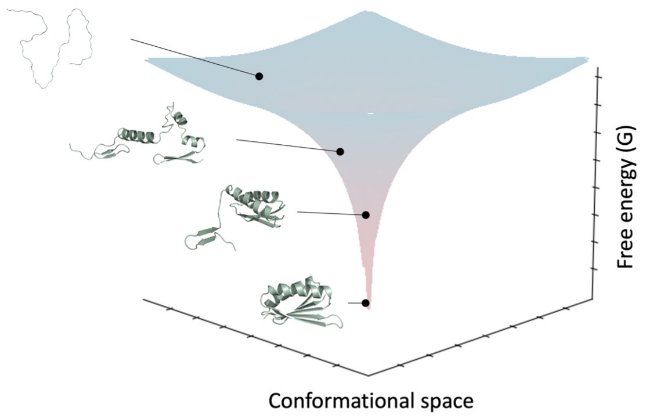

Figure 2: Hypothetical simplified free energy landscape for a folding protein.
A protein (in green) is folding from an unfolded state to its native folded state. Four snapshots of the folding process is captured, and the conformation (configuration in the more general case) is mapped onto the free energy surface. The protein randomly explores the free energy landscape but is more likely to visit states with lower free energy. Thus eventually, the protein will fold to its native structure, the conformation with the lowest free energy shown at the bottom.

This shows the free energy as a function of all the possible conformations of the protein. The protein is more likely to be found in a state of lower free energy. Thus, folding can be seen as a diffusion-like process where the protein randomly explores different nearby states but with a higher likelihood of going towards states with lower free energy. Eventually, the protein will find itself at the bottom of the free energy landscape with the lowest free energy in its native conformation.

The probability P(j) of finding a protein in a state j out of all conformations (from *i* to n) is given by the Boltzmann distribution:

$$P(j) = \frac{1}{Z} e^{(-E_j / kT)} \qquad Z = \sum_{i}^{n} e^{(-E_i / kT)} \tag{1.2}$$

The left-side equation is the Boltzmann distribution and Z is called the partition function. E_j is the energy of the state *j*, T is the temperature and k the Boltzmann constant.

1.3 Structural Classification

Now that we understand the basics of protein structure and folding, we can examine their structural classification. Protein structures are commonly divided into four subdivisions: *primary*, *secondary*, *tertiary*, and *quaternary* structure *(***Fig. 3***)*. The primary structure is the sequence of amino acids as it occurs in the protein. The secondary structure is the local 3-dimensonal arrangement of the backbone. The hydrogen-bonding capabilities of the backbone allow two common secondary structure elements to form, called the *alpha-helix* (α-helix) and the *beta-sheet* (β-sheet). These elements are so common that they are often depicted as coiled ribbons and arrows respectively while other parts of the protein are not. The next level higher up is the structure of the whole protein chain and this is called the tertiary structure. If a protein binds to other proteins, we call it a protein complex, or protein assembly, and the structure is referred to as the quaternary structure. A single chain in a protein complex is called a subunit. If a protein complex consists exclusively of proteins with identical sequences, we call it a *homomer* while if it exists of different proteins with different sequences, we call it a *heteromer*.

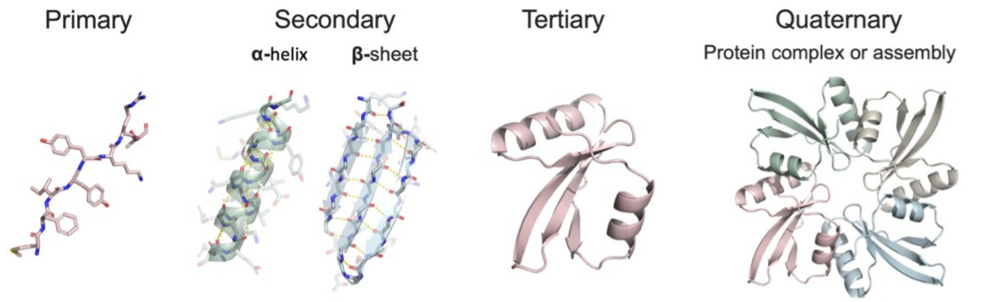

Figure 3: The four subdivisions of protein structure.
The primary structure is the sequence of amino acids as they occur in the protein. The secondary structure is the spatial organization of the backbone. Two common secondary structure elements are created by backbone hydrogen bonds (shown in yellow as dashed lines) and are called the α-helix and the β-sheet. The folding of the entire protein is called the tertiary structure. The entire structure of multiple proteins associated in a protein complex, or protein assembly, is called the quaternary structure.

Many databases exist to both store and classify protein structures. The Protein Data Bank (PDB)[2] is a database that stores most of the protein structures that have been resolved with a variety of techniques such as X-ray crystallography and Cryo-electron microscopy. Other databases store the full structures or parts of the structure based on further categorizations. In **Paper III** we use the three classification databases: CATH[3], SCOPe[4], and Fuzzle[5]. CATH and SCOPE are *domain* and *fold* databases. Domains are segments of protein structures that can fold on their own, and folds refer to the connections of secondary structure elements which can be either the full tertiary structure or parts of it. Fuzzle is a fragment database that stores protein fragments which are evolutionarily related. While the PDB consist of hundreds of thousands of proteins, the three other databases mentioned consist of many millions of protein domains, folds, and fragments between them.

1.4 Protein Symmetry

Another way to classify proteins is by their symmetry. Many protein complexes display symmetry and almost all homomeric protein complexes are symmetrical[6]. A protein complex is said to be symmetrical if there exists one or more transformations, such as rotations, that can map it onto itself without changing the structure. Symmetrical protein complexes can be divided into symmetry groups depending on which transformations can be applied. The most common protein complexes seen in nature are described by rotation transformations around a single point. These are also said to have point symmetry. The simplest symmetry group of this kind is the cyclical group (C_n), which has one or more rotation transformations about a single axis. The dihedral group (D_n) has one or more rotation transformations about one axis, and another 180-degree rotation perpendicular to the first axis. The cubic group is the most complex class with point symmetry, as they have two to three distinct rotation transformations at multiple points in space **(Fig. 4)**. Complexes with tetrahedral (T) symmetry have 2- and 3-fold rotation axes, complexes with octahedral (O) symmetry have 2-, 3- and 4-fold rotation axes, and complexes with icosahedral (I) symmetry have 2-, 3- and 5-fold rotation axes. Protein assemblies with cubic symmetry primarily have roles as containers for storage and transport[7]. Examples include ferritin, a protein container for iron storage, found inside almost all living organisms, including humans[8]. Another example is viral capsids, which are protein containers encapsulating the viral genome. Viral capsids will be explored further in the next section. Outside of the point symmetry group are symmetries with additional translation transformations such as helical and crystal symmetries.

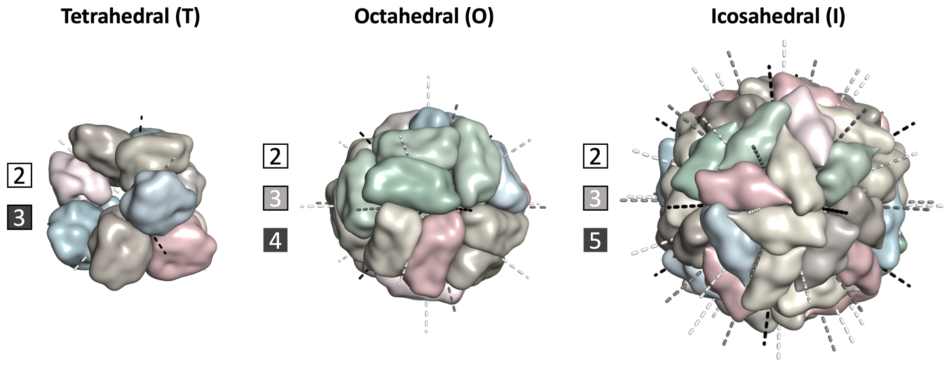

Figure 4: The cubic symmetry group.
Examples of protein assemblies with Tetrahedral (T), Octahedral (O) and Icosahedral (I) symmetry. Dashed lines indicate the different rotation axes, and the color indicates their rotation fold which are also highlighted by the box to the left of them. As an example, a 5-fold rotation means that along that axis the structure can be rotated 360/5 = 72 degrees and produce the same structure again. T symmetry contains: 2- and 3-fold rotation axes, O symmetry: 2-, 3- and 4-fold rotation axes and I symmetry: 2-, 3- and 5-fold rotation axes.

1.5 Viruses

Viruses are very small entities containing genetic material, and they infect and replicate inside other organisms from plants to bacteria and humans. They are responsible for a large number of human diseases and can cause large-scale pandemics, shutting down parts of the global economy and killing millions of people. The latest Covid-19 pandemic for instance has, as of the 28th of September 2023, killed almost 7 million people worldwide[9]. It is of significant interest to understand the structure and assembly mechanism of viruses to design drugs that can combat them. An important part of most viruses is a protein assembly, called a capsid, that encapsulates and protects its genetic material. The formation of the capsid is an important step in the formation of a virus, as the genetic material is either encapsulated inside the capsid as it forms or pumped into an empty capsid by molecular motors[10]. Viral capsids play a significant role in this thesis, as we have developed a method to predict their structure given their sequence in **Paper I,** and we analysed their structure and assembly mechanism in **Paper II**. Moreover, capsids are natural containers that could be used in many applications from drug delivery to vaccine design. In **Paper III** we developed a method to design capsid-like containers called cages.

1.6 Capsid Structure

Capsids most commonly have icosahedral symmetry and to a lesser extent helical symmetry[10, 11]. Caspar-Klug theory[12] describes how icosahedral capsids can be constructed with varying sizes. An icosahedral capsid can be thought to exist of a combination of even-sided hexameric triangles and pentameric triangles with 3 protein subunits inside each triangle. If both sides of all triangles must touch the geometry of triangles arranged in a hexamer is flat and the geometry of triangles arranged in a pentamer is curved (**Fig 5a**). To form an enclosed, 3-dimensional, icosahedral structure, 12 pentameric triangles must be connected, either directly or through any number of hexameric triangles. The size of the icosahedral capsid and the number of subunits (remember 3 for each triangle) then comes down to the spacing between the pentamers with the hexamers. In Caspar-Klug theory this can be more mathematically rigidified by illustrating it on a hexagonal lattice in two dimensions which are called **k** and **h** (**Fig. 5b**). The spacing between pentamers is the distance covered along k and h in hexagonal space. The distance is also called the triangulation number (T-number) and is mathematically defined as:

$$\text{T-number: } h^2 + hk + k^2 \qquad (1.3)$$

Virus capsids are grouped according to their T-number and are named accordingly, such as T1 (T=1), T2 (T=2), T3 (T=3) etc. **Figure 5c** shows some examples for T1, T3 and T4 and how the triangulation number relates the T-number to the spacing between pentamers and the 3-dimensional structure. The number of subunits in an icosahedral capsid is 60 times the T-number. The simplest capsids, T1, thus consist of only 60 subunits while some of the largest, such as the capsids found inside the Mimivirus are estimated to have up to 72000 (calculated from T=1200) subunits[13]. Capsids can be both homomeric and heteromeric. Subunits in homomeric T1 capsids can be completely symmetrical as each subunit is bound in a pentagonic fashion, while for larger T-numbers the same subunit has to bind both in a hexagonal and pentameric fashion and therefore cannot be. This principle where the same subunit binds in different fashions is called quasi-equivalence. Heteromeric capsids do not necessarily have quasi-equivalence as different subunits can accommodate either the pentameric or hexameric conformation. The ability of a heteromer to arrange into an approximate symmetrical configuration is called pseudosymmetry.

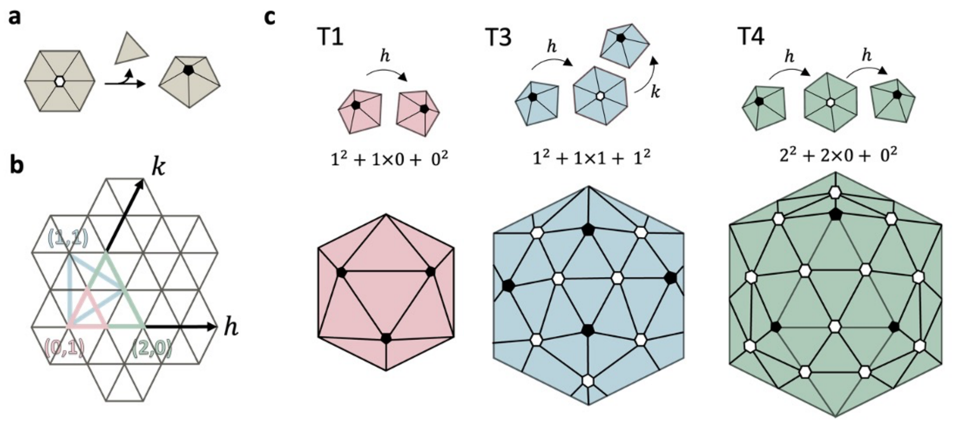

Figure 5: Caspar-Klug theory for classification of viruses.
a: 6 triangles arranged in a hexamer is geometrically flat. If 1 triangle is removed so the 5 triangles form a pentamer, the structure is geometrically curved. A hexamer is indicated by a white hexagonal center and pentamers are indicated by a black pentamer in the middle for all items present. **b**: Illustration of how the T-number can be calculated, and how it relates to the capsid structure. Different triangles can be created by moving along the k or h axis in hexagonal space. The vertices of the triangle for a given h and k are changed from hexamers into pentamers and 20 of those pentamers are connected to create the icosahedral capsid. Here and example is shown for T=1 (h=1, k=0), T=3 (h=1, k=1) and T4 (h=2, k=0). **c**: A visual representation of the figure in b. which also shows the 3-dimensional structure. Here T1, T3 and T4 are shown and how they are created by connecting pentamers through hexamers. The colours used to create the triangles in b matches the colours of the capsids in c.

2 Computational Modelling of Proteins

Broadly speaking there are two approaches to study proteins. The first is experimental where the protein of interest is measured and investigated in the lab using a plethora of different techniques. The protein is said to be studied *in vitro* ('in glass') or *in vivo* ('In the living') if it is in a test tube or in its biological context, respectively. The second approach is to simulate the protein of interest on the computer, which is then appropriately called *in silico,* as computer chips are made of silicon. In this chapter, we will cover some of the computational methods used for modelling proteins, with a focus on methods concerning the problems of predicting protein structures and designing them. Both problems can be phrased in terms of an optimization problem where the parameters of the protein (for instance the torsion angles of the backbone or the sequence) need to be optimized. In general, there are two broad approaches to finding the optimum. One is based on conventional sampling approaches guided by a score function and the newer deep learning (DL) methods that learn from data. In **Paper I** and **III** we use the macromolecular modelling software called Rosetta[14] as part of the structure prediction and design tools we have developed. Thus, there is an additional focus on how these methods are implemented in Rosetta.

2.1 Protein Representations

First, we need to understand how proteins can be represented *in silico*. Protein structures can be represented at different levels of detail depending on the modelling scenario and the question at hand. The true physical description of a protein would be a quantum mechanical (QM) description, but such descriptions are very computationally expensive and are usually intractable for structure prediction and design.

2.1.1 All-atom Models

A common description of the protein structure is an all-atom representation where the detail of the electronic structure is ignored, and atoms are instead represented by van der Waals spheres. This description allows the forces of the protein to be described by classic mechanical force fields such as is commonly done in molecular mechanics (MM). The advantage of such a representation is that all atoms are explicitly modelled, giving a more accurate description of the protein, but the disadvantage is that this can in some cases be too computationally expensive.

2.1.2 Coarse graining Models

To reduce computational expense, several atoms can be combined into a single sphere, averaging their physical characteristic. This representation is set to be coarse-grained. In Rosetta, for instance, a common strategy is to average the sidechains into what is called a centroid and keep the backbone in all-atom mode.

2.1.3 Solvation Models

Proteins are often solvated in water and the effect of water is very important to model. Explicit solvation models represent the individual water molecules in the simulations themselves. For instance, this is common in molecular dynamic (MD) simulations as described later. Implicit solvation models do not represent the molecules themselves but represent them implicitly through a continuous medium.

2.1.4 Tensor Representation

So far, we have described the models as used in many different simulation software programs. However, for DL-based methods, the protein must be represented as a tensor, which is an n-dimensional vector such as a 1D or 3D array. To transform the protein into a tensor, we extract measurable quantities of the protein called features. Some common features include:

- The protein sequence.
- Co-evolutionary information extracted from homologous sequences.
- Structural information extracted from homologous proteins of known structures.
- Physical contacts in the form of a contact map.

2.2 Score Functions

Now that we understand protein representation, we need to get back to the optimization problem. As we navigate the parameter landscape in search of an optimal solution, we need a guide that tells us how well our current parameters satisfy the solution we are looking for. Specifically, we need a function that takes the current values of the parameters and tells us if the solution is better or worse, relative to other parameter values. For structure prediction and design, we are interested in a function that can map the parameters onto the energy or free energy as described in Chapter 1. Such a function is commonly called an energy function or just a score function. It often includes several terms that together approximate the real energy, with parameters that are trained on experimental data to recapitulate real energies.

2.2.1 Force Fields

Force fields describe the forces or potential energies between atoms in a system. Molecular Mechanics (MM) force fields use classical mechanical, physics-based representations, where the energies are divided into bonded terms and non-bonded energy terms depending on whether they are covalently linked or not. These terms are then summed up to give the total energy of the system. The basic form of the bonded terms can include bond stretching, bending and torsion:

$$V_{bonded} = \sum_{bonds} k_b(b - b_o)^2 + \sum_{angles} k_\theta(\theta - \theta_o)^2 + \sum_{torsions} k_\phi[1 + cos(n\phi - \delta)]^2 \quad (2.1)$$

The basic non-bonded terms usually include the electrostatic and van der Waals interactions modelled through the Coulomb force and Lennard Jones potential, respectively as:

$$V_{bonded} = \sum_{i<j} \frac{q_i q_j}{4\pi D r_{ij}} + \sum_{i<j} [\frac{A_{ij}}{{r_{ij}}^{12}} - \frac{C_{ij}}{{r_{ij}}^{6}}] \tag{2.2}$$

2.2.2 Knowledge-based or Statistical Potentials

Observed frequencies of states of proteins, such as torsion angles of the backbone, as derived from the available structural databases such as the PDB, can be used to derive energies. As explained in Chapter 1. on protein folding, there exists a relationship between the energy of a particular state, and its probability of observing it given by the Boltzmann equation (eq. 1.2). Inverting this relationship means you can derive energies from these probabilities, and these energies are referred to as knowledge-based or statistical potentials.

2.2.3 The Score Function in Rosetta

The score function in Rosetta is derived from a combination of knowledge-based terms and physics-based terms. The physics-based terms include variants of the Coulomb and Lennard Jones energies described earlier, and the statistical terms include energies for rotamers and backbones. It furthermore uses an implicit solvation model.

2.3 Search Methods

If we could enumerate all possible parameter combinations and calculate their associated score, we could just pick out the best and we would be done. However, there's an astronomically large number of parameters to sift through, and such brute-force methods are rarely useful. Instead, we need to utilize search algorithms that can smartly navigate the energy landscape to avoid the need to visit every single point. If we were to map the energy function onto the parameters to visualize the energy landscape as was done in **Figure 2**, it's usually much more complicated and filled with peaks and valleys. When trying out different parameter configurations, it's easy to think you've found the best solution, as there are no other better solutions in its vicinity. We refer to these false best solutions as a local minimum, and the true best solution as the global minimum. The goal of search algorithms is to find the global minimum in the most efficient way possible. In practice, however, it is often very hard to find the actual global minimum, and therefore we often content ourselves with a satisfactory solution. In the following sections, we will explore some of these search methods

2.4 Gradient descent Minimization

As the parameter search is ongoing, it is often useful to find the local minima around the current parameter configurations. Gradient descent minimization is a method with many flavours that can be used to find a local minimum. The general idea is to calculate the first partial derivative or gradient $\nabla E(\Theta_i)$ of the energy function at the current point (Θ_i), and then take a step of a given size ε along the gradient to go a new point (Θ_{i+1}). This can be expressed as:

$$\Theta_{i+1} = \Theta_i - \varepsilon \nabla E(\Theta_i) \tag{2.3}$$

When we have reached Θ_{i+1} a new gradient is calculated (∇E_{i+1}) and a new step is taken along the gradient. The search iteratively continues until the gradient is changing very little, aka we have reached a local minimum. The approach can be improved by utilizing the second partial derivate, called the Hessian, as it gives more information about the curvature at Θ_i. Quasi-Newton methods use parts of the Hessian, as it is computationally expensive to calculate, and these methods are very popular gradient descent algorithms. Rosetta uses the quasi-newton-based gradient descent Broyden-Fletcher-Goldfarb-Shanno (BFGS)[15] implementation by default.

2.5 Monte Carlo Simulations

The problem with gradient descent is that it ends once a minimum has been found. It's unlikely that the first minimum we find is the global one, and therefore we need algorithms that can 'jump' out of local minima and explore different parts of the search space. Monte Carlo[16] (MC) simulation is one such approach to do so. It relies on making a stochastic jump in the parameter space and evaluate if such a move should be accepted or reverted to the previous state. How do we decide when to accept a move? As discussed in Chapter 1. about protein folding, the probability of seeing a particular protein state is proportional to the Boltzmann factor (eq. 1.2). We therefore choose to accept a move based on the Boltzmann factor, by calculating the energy of the two states. If the energy is lowered, the Boltzmann factor says it is a more likely state and we accept it. If the energy is higher, we sometimes still want to accept. Remember, we want to be able to 'jump' out of local minima and discover new areas of parameter space. If the energy is higher when going from state *i* with energy E_i to state j with energy E_j the probability of accepting the move is proportional to the difference in the Boltzmann factor between the two states:

$$P(j) = e^{-(E_j - E_i)/kT} \tag{2.4}$$

Where k is the Boltzmann constant and T the temperature per usual. This algorithm is also called the Metropolis-Hasting algorithm[17] and the accept criterion the Metropolis criterion. MC simulation is often combined with simulated annealing[18]. Notice the temperature factor in eq. 2.4. In simulated annealing, the temperature starts high and is subsequently lowered (annealed) as time goes on. At a high temperature it is more likely higher energy states will be selected while for lower temperature it is the opposite. Thus, using simulated annealing, a larger space can be initially explored while later exploration focusses on lower energy or more probable parameters. MC methods are used in many modelling software packages and many protocols inside Rosetta uses the MC simulated annealing method.

2.6 Evolutionary Algorithms

MC simulations are often run independently but it can be advantageous to have different trajectories learn from each other to collectively home in on the global minimum. Evolutionary algorithms (EA) are a class of algorithms that simulates a population of different trajectories (called individuals) that share information during the search. They do this by different biological evolution inspired operators such as *mutations* and *selection*. The score function in an EA algorithm is often referred to as a fitness function as it evaluates the fitness of each individual to determine which ones continue in the population.

2.6.1 Differential Evolution

EAs have many variants, and one variant called differential evolution (DE) is particularly useful when solutions are real-value numbers. The basics of the DE algorithm are explained in the following.

Let us define a population to contain *L* individuals and operate over *G* generations. Each individual *i* in the population contains an n-dimensional vector of size N. Let us call this vector $x_{i,g}$. We thus have the following definitions for an individual *i* with the vector $x_{i,g}$ as:

$$x_{i,g} = [n_1, n_2, n_3 \dots n_N], \qquad i = 1,2,3 \dots L, \qquad g = 1,2,3 \dots G \tag{2.5}$$

The elements in the $x_{i,g}$ vector can represent many things such as the degrees of freedom of the protein being modelled.

The DE algorithm progresses iteratively over *G* generations, applying two operators, mutation*,* and crossover, to each individual *i* at every generation *g* to produce a trial vector. This trial vector is then either accepted or rejected based on a selection criterion.

2.6.1.1 Mutation

A mutation vector ($m_{i,g+1}$) is generated by combing 3 other individuals in the population that does not include $x_{i,g}$:

$$m_{i,g+1} = x_{r1,g} + F \times (x_{r2,g} - x_{r3,g}) \tag{2.6}$$

Where F > 0 (called the *differential weight)*, and r1, r2, r3 being randomly chosen vectors from the population (not including *i*).

2.6.1.2 Crossover

The trial vector ($t_{i,g+1}$) is created by randomly combing the elements $m_{i,g+1}$ and $x_{i,g}$ based on the crossover probability (CR), where CR ∈ [0,1]. For each element j in $m_{ij,g+1}$, a random number is calculated between 0 and 1 and the jth element is taken from either $m_{ij,g+1}$ or $x_{ij,g}$ depending on the condition:

$$t_{ij,g+1} = \begin{cases} m_{ij,g+1} & if\ rnd[0,1] \leq CR \\ x_{ij,g} & otherwise \end{cases} \tag{2.7}$$

One random element however is always picked from $m_{ij,g+1}$ regardless of the value of $rnd[0,1]$.

2.6.1.3 Selection

Finally, a selection criterion is used to pick either the trial vector $t_{i,g+1}$ or $x_{i,g}$ to continue into the next generation g+1. A common selection criterion to use is the greedy criterion, which means that the best out of the two, according to the fitness function, is the one that is picked.

This process of mutation, crossover and selection happens to all individuals in the population over *G* generations. The final solution is the individual with the best fitness value.

2.6.2 Memetic Algorithms

Memetic algorithms are an extension of EAs that apply a local search strategy to improve the results of the solutions imposed at each generation. A local search strategy is any strategy that improves upon the current solution by finding a nearby one in the parameter space that is better. Such a strategy can be for instance be a MC algorithm or gradient descent. Imagine once again a free energy landscape such as in **Figure 2** with many peaks and valleys. A trial vector created by the DE mutation and crossover operators is not guaranteed to produce the best result in its vicinity. Potentially good solutions can be missed if the surroundings are not explored, and therefore memetic algorithms are powerful hybrid approaches, that take advantage of both exploiting the local landscape and exploring the global landscape.

2.7 Molecular Dynamics

Molecular dynamics (MD) methods use a force field to evolve a system over time in discrete time steps. At every time step, the atoms are moved according to the forces that act on them as described by the force field. The time step is chosen to be shorter than the fastest molecular motions in the system, which is usually the bond vibrations (which are in the order of pico- to femto-seconds). While MD is extremely useful for many different modelling scenarios it can be very computationally expensive to model proteins for structure prediction and design objectives.

2.8 Symmetry Approximations in Rosetta

In general, the more amino acids that are modelled, the more computationally expensive the calculations of the total score will be. As discussed in Chapter 1. on protein symmetry, many protein complexes are symmetrical. Using this feature explicitly in the energy calculations can help reduce the computational expenditure when modelling large systems. There are two reasons for this:

1. Only unique interactions need to be calculated.
2. Only atoms with unique interactions need to be represented.

Consider a symmetrical tetramer (a protein complex of 4 chains) in **Figures 6a** and **6b**. In **Fig 6a**, without considering symmetry, to calculate the total energy one needs to consider the internal energy of all four chains, as well as all interactions between all chains. In **Fig 6b**, with symmetry considerations, to calculate the total energy one needs to only consider the internal energy of one chain and its unique interactions only, significantly reducing the computational complexity. The full energy can be calculated by multiplying the unique energies in appropriate ratios. For the example in **Fig 6b** the full energy is 4 x the internal energy of chain 1 plus 4 x 1:2 interaction plus 2 x 1:3 interaction**.** If symmetry is not considered, the problem is only exacerbated for larger systems. For an icosahedral protein capsid as shown in **Fig 6c**, calculating the total energy, without modelling symmetry, would mean calculating the internal energy of at least 60 chains, as well as all their interactions - clearly an intractable computational problem. However, considering only the unique chain and its unique partners (see **Fig 6c**) makes modelling of such large systems possible. This symmetry representation is possible in Rosetta[19].

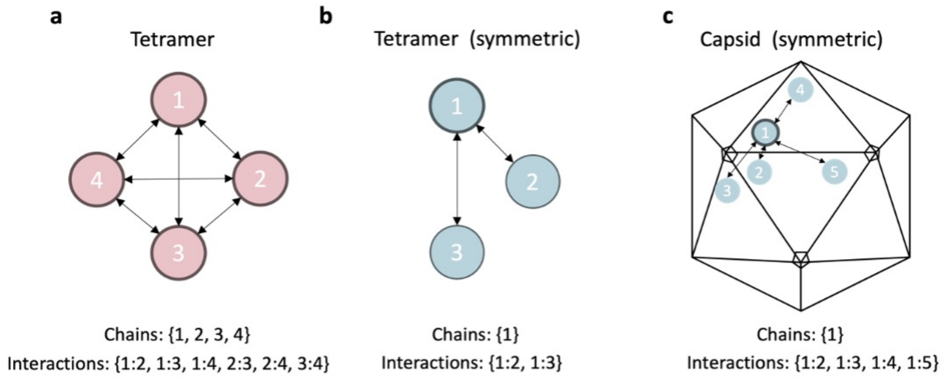

Figure 6: Modelling protein complexes with and without symmetry
a: A symmetric tetramer without explicit symmetry modelling. Proteins are shown as circles. Internal energies that need to be considered within the chains are highlighted with bold circumference and interactions between chains that need consideration are highlighted with arrows. The total set of considered chains and interactions is shown below the figure. **b**: The same tetramer as in a, but modelled with symmetry. The computational complexity of calculating the total energy is reduced, as only 1 internal energy (chain 1) needs to be calculated, as well as only 2 unique interactions. **c**) Modelling of an icosahedral T=1 capsid containing 60 chains with symmetry. The case is similar for the tetramer in b but more unique interactions are present.

2.9 Deep Learning

Deep learning (DL) is a branch of machine learning and artificial intelligence (AI) that utilizes artificial neural networks (NN) to learn from data to accomplish various tasks. DL-based methods have been behind the recent success of AI in the previous years: from image, video and audio processing[20], to powering advanced language models such as ChatGPT[21] and gaming AIs capable of defeating world-class players[22]. Recently, they also started to outcompete previous methods in both protein structure prediction and protein design, but this discussion is saved for the upcoming chapters. Here we explain some of the basics behind DL as it pertains to protein structure prediction and protein design.

2.9.1 A Simple Neural Network

The artificial NN of DL models are inspired by the brain's neural circuitry, where information is transferred from neuron to neuron through the firing of chemical signals. A simple NN is shown in **Figure 7a.** The nodes represent neurons, and the connections between nodes are neural connections. Data in the form of a tensor is fed into the NN from the left, and information travels through the NN to the right. The NN contains layers of connected nodes, and each node contains weights, a bias and an activation function. The weights and biases are the parameters of the DL model and the values of these are learned during training. A node combines the activation function, the learned weight, and its input values, to produce a new value that is then fed into the next nodes. In this way the nodes learn to fire (produce an output value), akin to the way biological neurons fire through chemical signals.

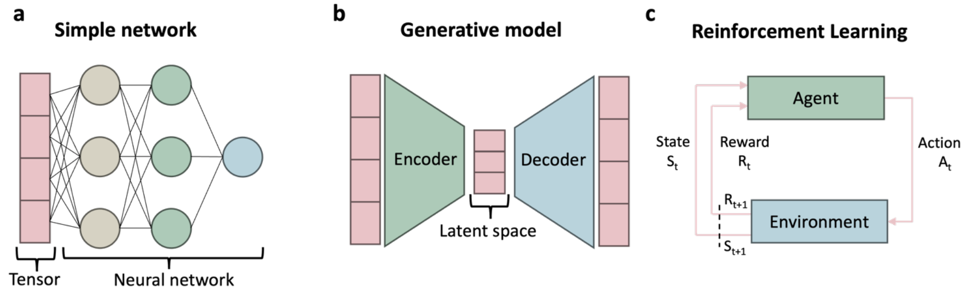

Figure 7: Deep learning models.
a: A simple neural network, also called a multilayer perceptron (MLP), consisting of 3 layers. The input data, in the form of a tensor, is connected to a 3-layer neural network (shown in different colours). **b**: An example of a generative model in the form of a variational autoencoder (VAE). An encoder encodes training data into a low-dimensional vector called the latent space, consisting of probability distributions that represent the data. A decoder learns to sample this probability distribution to reproduce the data. After training, the encoder can be removed and the latent space plus the decoder can be used to generate (or mimic) new data. **c**: A general reinforcement learning loop. An agent in a state (S_t) and associated reward (R_t) takes an action (A_t) in an environment that changes its state (S_{t+1}) and gives it an associated reward (R_{t+1}). The model ultimately learns a policy: the actions to take to maximize its total reward.

A DL model can learn its parameters (weights and biases) in different ways. In supervised and unsupervised learning, training happens through either labelled or unlabelled training data. In the former, the DL model is trained to predict the correct label from the training set. The model is presented with a labelled training set and adjusts the weights and biases until it can predict the labels as well as possible.

The adjustment of the weights happens through a process called backpropagation. The error in the prediction is expressed mathematically in terms of a loss function. The goal of backpropagation is to minimize the error in the loss function, by updating the weights and biases in the model through a gradient descent algorithm[23].

Supervised learning has become important in fields such as protein structure prediction, where the model can be fed datasets of sequences and proteins of known structures to learn to predict from.

2.9.2 Generative Models

In unsupervised learning, we also present a large dataset to the model, but this time it is not labelled, and the model must find meaning on its own. Note, that the distribution of layers, connections, choice of activation functions, loss functions and so on are referred to as the architecture. The simple architecture in **Figure 7a** is also called a multilayer perceptron (MLP). **Figure 7b** shows another DL architecture called a variational autoencoder (VAE). It looks different, but the underlying layout is still a NN. It consists of an encoder and a decoder. The encoder (a NN) learns to encode the training data into a lower dimensional vector space called the latent space. The latent space consists of probability distributions, specifically encoded into 2 numbers: a mean and standard deviation. A decoder (also an NN) learns to decode samples from the latent space to reconstruct what was encoded. The loss function for this architecture is two-fold: one that ensures as little reconstruction loss as possible between the encoded and decoded data and one that ensures the probability distributions are smooth and regular. Since VAEs just learn to reconstruct the training data, it is considered an unsupervised or self-supervised model.

Importantly, VAEs are part of a larger class of generative models that can mimic new versions of the training data. VAEs can do this by sampling from the learned encoded latent space. Generative models have become important for protein designs for instance, as variants and even unnatural proteins can be generated from such models.

2.9.3 Reinforcement Learning

Another way that DL models can learn is through reinforcement learning as shown in **Figure 7c**. In reinforcement learning (RL), an agent (which could be a protein) takes an action in an environment (which could be 3D space). The action changes the agent's state and given the agent's new state, a reward, good or bad, is received. By learning which actions maximize its total reward, the agent can learn to accomplish its task as well as possible. The decision-making in what actions to take to maximize its reward is called its policy, and the policy is what is continually updated during training. This way of learning is akin to how humans learn, as we also make decisions based on what has rewarded us (in general of course.)

3 Protein Structure Prediction

Protein structures can be determined experimentally by techniques such as X-ray crystallography, Nuclear Magnetic Resonance (NMR) spectroscopy and Cryo-electron microscopy (Cryo-EM) but the process can often be laborious, time-consuming, expensive, or impossible owing to the difficulty of expressing and solubilizing certain proteins or in the various ways they need to be prepared. Computational structure prediction offers a less laborious, faster, and less expensive route to predicting protein structures but has remained a long-standing challenge of science[24] owning to its inherit difficulty[25] and its great potential for transforming the medicinal and biotech industry[26]. Recently, the problem of protein folding has been claimed to be solved with the advent of the deep learning model AlphaFold[27] from Google's DeepMind group. In this chapter, I will describe different methods for protein structure prediction, starting from *ab initio* methods, and work our way up to the new DL-based models such as AlphaFold. A brief account of the history of structure prediction will be given through the CASP and CAPRI competitions. I will discuss the current limitations of AlphaFold and its protein complex prediction counterpart AlphaFold-Multimer[28] in detail. Finally, I will discuss the prediction of protein assemblies and how we overcome some of the challenges of predicting large protein assemblies, as is done in **Paper I**.

3.1 Pure *Ab Initio* Methods

Ab initio is Latin and means "from first principles" and originally refers to modelling approaches where only the physical laws of nature are used. Nevertheless, nowadays, *ab initio* methods are used as an umbrella term that refers to methods that predict structure from sequence without starting from templates (template modelling is described later). In this section I will describe pure physics-based *ab initio* methods, and in the following section, *ab initio* methods that rely on database information. Pure *ab initio* methods commonly use an energy function in an all-atom representation, where the parameters are sampled by either MC or MD methods. The main advantage of pure *ab initio* methods is their ability to discover new folds and simulate folding without assuming knowledge of the underlying structure[29]. Nonetheless, while *ab initio* method has had success in predicting the structure of small proteins[30], their extension to larger proteins remains challenging because of the large conformational space available[29, 31]. All-atom and energetic-based methods are however still important for local optimization, which occurs in the latter stages of prediction, where the model must be refined.

3.2 *Ab Initio* Methods with Database Information.

Data-driven models that build on existing sequence or structural database information have been a key driving factor in the field. This has been facilitated by the growing numbers of experimentally determined protein structures and available genome sequences over the years. Pure physics-based *ab initio* methods as described previously, can be improved by utilizing database information. Such database-based *ab initio* methods are often interchangeably called *de novo* methods, meaning 'from the beginning'.

Major improvements in predicting protein structures have come from assembling them from smaller backbone fragments, usually 3-15 residues long. Using fragments in the modelling, a strategy also referred to as fragment assembly, is a strategy that has helped reduce the conformational space to search through. The concept is based upon the connection between the local structure of the backbone ($\psi,\ \varphi$) and the local sequence[32]. Instead of sampling the backbone coordinates directly, a library of fragments, based on the local sequence, can be inserted directly into the backbone.

Major improvements derived from sequence databases have come from using coevolutionary information[33]. The concept is based on the idea that amino acids that are close in 3D space and interact are likely to mutate together. To find information on these covarying mutation patterns, the sequence of the protein of interest is matched to other similar sequences in a multiple sequence alignment (MSA), and the pattern can be extracted by looking for these changes in the MSA. A contact map is usually created from this which can then either be used for constraints or features in a DL model for instance.

3.3 Template-based Methods

With the growing number of experimentally determined protein structures over the years, of which are now in the hundreds of thousands, it has become more and more common to use existing templates to aid modelling. These approaches are called template-based methods of which there are two main approaches.

3.3.1 Homology Modelling

In the case where there is enough sequence homology (usually 30%) of the query sequence to another sequence with a known structure, homology modelling has traditionally been used[34]. Homology modelling is based on the idea that proteins that have similar sequences and are evolutionary related have similar structures. In homology modelling, a sequence alignment is created to find matching sequences to a query sequence. Aligned regions are then used to create an initial backbone structure, and unaligned regions, loops, and sidechains are then modelled, followed by a final energy refinement[26, 29].

3.3.2 Fold Recognition

In the case where there are no homologous templates available, another approach called fold recognition (or threading) can be used[35]. Fold recognition is based on the idea that that structure is more conserved than sequence, and that there are a limited number of protein folds in nature[36, 37]. In fold recognition, the query sequence is threaded onto a library of folds and the best match is found by evaluating it in the context of a knowledge-based score function[26, 29, 38].

3.4 Protein Docking

So far, we have discussed the prediction of single-chain proteins and protein complexes under the same umbrella. However, protein complex prediction comes with additional challenges as one must not only predict the structure of the individual subunits but also the relative position of the subunits with respect to each other. Given this challenge, the subunits of the complex are often considered rigid bodies, and the main challenge is to find the best

6-dimensional parameters (3 translations and 3 rotations) of the subunits. The most successful strategies have been the use of Fast Fourier Transform (FFT)-based methods[39]. In FFT methods, the complex is sampled on a grid and docking models with good metrics such as shape complementarity and energetics are found[40]. Commonly visited FFT-based docking programs include for instance ClusPro[41] and ZDOCK[42]. The advantage of FFT-based methods is their speed, and they are often the first line of attack for complex prediction to find putative binding sites[43]. The main problem with FFT-based methods however is their lack of accountability for backbone and side chain flexibility[44]. By allowing some atomic overlap, also called soft docking, in conjunction with energetic refinement (such as with CHARMM[45] in ClusPro) this can be mitigated to some extent[40]. Other methods allow flexible backbones throughout the protocol or for last-stage refinement through MD, MC or normal mode analysis (NMA)[43]. One way to incorporate flexible backbones, inspired by the conformer selection model[46] is ensemble docking. In ensemble docking, different backbones are used throughout the sampling with the goal of the native being selected in the end. One example of this is RosettaDock which compiles backbones based on NMA, Backrup and Relax[47, 48]. Flexible backbone docking is still largely an unsolved field, and proteins undergoing large flexible changes remain challenging. Most docking protocols use coarse-graining in the first part of the protocol. Coarse-graining has the advantage of mitigating some of the computational complexity, and smoothing out the landscape, but can in some cases miss the native state, as it could be hidden by the coarse-graining model. The real docking landscape is more accurately represented by an all-atom model, but this has not previously been computationally tractable before the later stages of docking.

3.5 EvoDOCK

EvoDOCK is a protein docking software that only uses an all-atom representation of the protein. It uses a memetic algorithm, which combines a differential evolution algorithm for global exploration, and a local search based on MC and gradient descent optimization[49] (see also section 2.6 on evolutionary algorithms). EvoDOCK uses an ensemble docking-based strategy to sample different backbones, which are swapped out throughout the protocol, while it optimizes the 6 degrees of freedom describing the rigid body. In benchmarks, it showed improved accuracy and up to 35 times speed improvements over the most similar docking protocol RosettaDock 4.0 (However without the added motif dock score). It also showed similar accuracy to ClusPro, with better performance for some targets, albeit about twice as slow in general. Nonetheless, compared to ClusPro, it incorporates side chain and backbone flexibility in an all-atom mode throughout the protocol. We have extended EvoDOCK to model symmetrical protein complexes, to carry out the studies done in **Paper I-III**, and changes to EvoDOCK are highlighted in **Paper I.**

3.6 Structure Prediction Competitions: CASP and CAPRI

CASP (Critical Assessment of Structure Prediction) is a biannual protein structure prediction completion where research groups around the world attempt to make the best prediction for soon-to-be experimentally determined protein structures. The competition falls into two categories: groups that have three weeks to complete their predictions and online servers that must return their submissions within 72 hours. The competition started back in 1994 (CASP1) with 35 participating groups but has in 2022 (CASP15) grown to approximately 100 research groups. CASP divides its methods into mainly two categories; those that use templates, called Template-Based Models (TBM) and those that are template-free, called template Free Models (FM).

CASP has historically focused on the prediction of monomeric proteins and therefore CAPRI (Critical Assessment of Predicted Interactions) was initiated in 2001 for the prediction of protein complexes, with a similar competition setup to CASP. It runs its own competitions on a rolling basis but has merged its predictions into CASP when it rolls around biannually. Both CASP and CAPRI serve as the benchmark of the best-performing protein prediction models out there and are important events that gauge the current state of the field.

3.7 Deep Learning and AlphaFold

The account of the CASP competitions given here is based on the review by Wodak et al[40]. The first competitions CASP1 and CASP2 highlighted the challenge of using template-free models, with the only meaningful predictions coming from template-based methods for models with high homology to known structures. CASP3 and CASP4 saw improvements from better alignment tools, template usage and fragment assembly, but predicting structures with low homology remained challenging. In the following two decades, from CASP5 to CASP12, only moderate improvements were made, even with the introduction of co-evolutionary data in CASP10. However, in CASP13 DL models had matured to the point where a breakthrough occurred. The initial implementation of the DL model AlphaFold (AF), and other similar DL models, showed significant improvements in the ability to model targets in the FM category. AlphaFold also fared well in the TBM category even though it did not use templates[50]. AF in CASP13 used a NN architecture, that relied on using co-evolutionary data from MSAs, to build torsion-based constraints that could then be minimized with gradient descent to produce a final model. In the following competition CASP14, an even bigger breakthrough occurred as the successor of AlphaFold, AlphaFold2 (AF2), blew the competition out of the water in all categories, as it achieved close to experimental level prediction accuracy for many targets. Differing from its predecessor, it now included templates, an improved transformer-based NN architecture, and an end-to-end implementation. After the results of the competition were revealed, AlphaFold2 was hailed as the AI breakthrough in science, and many considered the long-standing challenge of predicting protein structure from sequence to be solved.

3.8 The Future of Protein Structure Prediction

Even though AF2 was a quantum leap in our ability to predict protein structures, the problem is far from solved. After the initial hype of AF2 and its implementation into the scientific community, it became apparent that many issues were still there. AF2 cannot predict all structures correctly. Most of the human proteome (98.5%) has now been predicted with AF2 but only about 36% of it can be predicted with high confidence[51, 52]. AF2 has not replaced experiments either, as they still serve as the gold standard for structure determination, while AF2 models mostly serve as hypothesis building[53]. It struggles to predict many classes of proteins such as proteins containing disordered regions such as loops[32] or intrinsically disordered proteins[54] and membrane proteins[55]. It is unable to predict non-protein interactions such as proteins with ligands, cofactors, metal ions, DNA/RNA and post-translational modifications such as glycosylation, phosphorylation and methylation[56]. It only predicts a single conformer and it struggles to distinguish between apo and holo forms[57] and the effect of point mutations[58]. Finally, it requires quite extensive resources both in terms of storage for the sequence and template libraries but also for hardware.

However, AF2 is continually being improved and its shortcomings are addressed, both internally by DeepMind (AF2 is now at v. 2.3) and externally by other researchers[59, 60, 61, 62]. It is still to date the state-of-the-art in structure prediction as no other method has so far matched its accuracy. In the most recent CASP (CASP15) all the best-performing models were based on AF2[63] (DeepMind did not participate). Nonetheless, other methods such as RoseTTAFold2[64] are catching up to AF2, and a new wave of exciting natural language processing (NLP) based methods such as ESMFold[65] are faster and can predict structures from single sequences.

So far in this history recap, I have left the very important discussion of protein complex prediction out of the picture. Predicting protein complexes is extremely important as most of protein functions are mediated either by transient or stable protein complexes[66]. DeepMind has recently launched AlphaFold-Multimer[28] (AFM) which to date is also the state-of-the-art in protein complex prediction. However, there is still room for improvement in particular for larger protein complexes as performance rapidly declines after complexes with more than two chains[67].

3.9 Protein Assemblies: The Next Frontier

Protein assemblies of multiple chains carry out many of the fundamental tasks of cellular functions. However, they are notoriously hard to predict because of their multiple subunits and their complexity. Advances have been made to address AFM's inability to predict larger protein structures. One approach is using sequential or combinatorial assembly approaches where subcomponents are predicted with AFM, and then sequentially connected by superposition (MolPC[67]) or transformations (CombFold[68]). MolPC for instance has proven to predict structures of up to 30 chains. ESMFold, RoseTTAFold2 and OpenFold[69] are other examples of protein complex predictors, but their current ability to predict larger assemblies has to the author’s knowledge not been demonstrated.

As addressed in **Paper I**, larger protein assemblies rely heavily on utilizing symmetry. A survey carried out in **Paper I** of protein assemblies in the PDB above 10 chains, revealed that 72% have global symmetry, 16% have local or pseudo symmetry, and only 12% are asymmetrical. Employing symmetry directly in the modelling of large protein assemblies is therefore very desirable. Efforts to model symmetry directly into AFM are being addressed[70], but currently with poorer accuracy compared to AFM. In **Paper I,** we show how large assemblies of the most complex symmetry types can be predicted with high accuracy. These constitute, to the author's knowledge, the method that can predict some of the largest assemblies from sequence, or *ab initio*, to date.

4 Protein Design

Natural proteins have evolved through biological evolution to carry out a particular function associated with the fitness of the organism. Humans have for a long time used these natural proteins, and repurposed them for our own benefit, for instance, to brew beer (thank you enzymes!). Nevertheless, repurposing natural proteins has its limits as evolution has only explored a fraction of the potential sequences and their usages. Protein design is a field that explores the realm beyond natural evolution, by either modifying existing proteins or creating new proteins from scratch. We already use modified proteins in our everyday lives. For instance, in our laundry detergents, we have enzymes that have been modified to work at different temperatures and pH, and in the textile industry, they are used to make our blue jeans look nice. Nonetheless, these are classic examples of modifying natural proteins. The design of completely new proteins, a field that is rapidly evolving, promises to completely revolutionize our lives through new medicines and novel applications.

4.1 The Protein Universe

It is helpful (and fun) to think about the vast space of different sequences we can explore for protein design as illustrated in **Figure 8**. Because evolution occurs by random mutation and natural selection, sequences tend to cluster around protein families (shown in brown), and this leaves a vast unexplored space of sequences available (shown in black). These unexplored regions of the protein sequence space have also been referred to as the 'the dark matter'[71] of proteins or the 'never born proteins'[72].

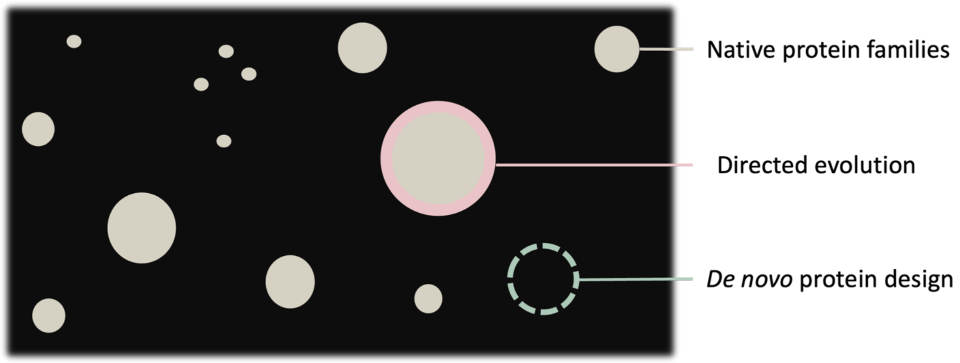

Figure 8: The protein universe.
The dark regions represent unexplored sequences and the clusters (brown) of existing native protein families. Native proteins, as evolved from evolution tend to cluster around protein families with similar sequences, folds and functions. Directed evolution starts from native proteins and can probe the surrounding universe (shown in pink). *De novo* protein design however does not need to start from existing protein sequences and can therefore explore any sequence in the protein universe. This illustration has been inspired by Huang et al[73].

As we gaze into the darkness, we can ask ourselves how many proteins can we possibly design? Well, a simple protein with a length of 100 residues can have 20^{100} different sequences, a number that is vastly more than the number of atoms in the observable universe[74]. So that is a lot. However, most designed sequences are not able to fold into the intended structure, express, or even be soluble which is why designing proteins is really challenging.

From humble beginnings, the field of protein design has come a long way, akin to the evolution of astronomy from rudimentary telescopes to sophisticated space missions. Today, the capacity to engineer an extensive array of new proteins stands as a testament to the monumental progress achieved as we will explore briefly.

4.2 Computational Protein Design

There are two ways to go about designing new proteins: experimental and computational. The most powerful experimental method for protein design is directed evolution[75]. Based upon the principles of natural evolution, a library of protein sequences goes through iterative rounds of mutation and screening for a particular desired trait. After several generations, this process can improve the performance of the initial protein, or even new functions can be obtained[76, 77]. Nonetheless, directed evolution requires starting with initial protein sequences and sequence exploration is often limited to the starting point (**Fig. 8**). Computational protein design (CPD) is done *in silico,* through smart algorithms and in some cases with the usage of high-performance computing. As the protein is designed *in silico* it doesn't require initial starting sequences and we are free to explore any sequence of our imagination. Regardless, directed evolution is still a very important technique, and is often combined with CPD to address its shortcomings, in particular for downstream optimization.

Before moving on, I will make two common distinctions between CPD methods. The first is perhaps better called protein redesign, as it deals with the modification of naturally existing proteins. A common goal is to either stabilize or increase the natural activity of a protein, but at the end of the day, the function of the protein is the same i.e., a better shovel is still a shovel. The second is *de novo* protein design. *De novo* is Latin and means 'a new' or 'from the beginning' and usually refers to the design of proteins with sequences, structures or functions not seen in nature before. The rest of the chapter will focus on CPD of proteins that have been designed *de novo.*

4.3 A Brief History of *De Novo* Protein Design

4.3.1 The Beginnings

Before the efforts of designing new proteins could begin the stage needed to be set. From the late 50s and onwards, high resolution structures of proteins were achieved[78] and in the 70's started to be catalogued in the PDB[79]. The ability to synthesize novel proteins came with the development of solid-state synthesis and later with gene synthesis. Computer power, along with the development of MM, MC and MD also needed to mature, as well as the understanding of protein folding and kinetics. Starting around the 80's the scene was set for the first attempts at designing new proteins.

Initial efforts were focused on using simple physiochemical principles combined with either bioinformatical knowledge and/or rudimentary computational algorithms. One early pioneering example is the design of the 4-helical bundle, *α4,* by Regan, Ho and Degrado et al[80, 81, 82, 83]. In a series of studies starting in 1987, the 4-helical bundle was designed using simple chemical heuristics of packing hydrophobic residues on the inside, and polar residues on the outside. Additionally, they utilized bioinformatically derived helix-promoting residues, and very simple algorithms that were limited to only a few polar/apolar residues. The simple structure of the helical bundle was used as a model system for protein design in the early years. Coiled coils, a subtype of helical bundles, were particularly studied, and still are to this day. The simple sequence-to-structure relationship (such as the design of hydrophobic (h) and polar (p) hpphppp-heptad repeats and packing of the highly defined knobs-into-holes interactions) means that they are very amenable to be designed. The massive effort into studying coiled coils means that the sequence-to-structure relationship is now largely understood[84].

Now in the 90s. Following Morse's law, computer power kept increasing and algorithmic design kept evolving, such as through the dead-end-elimination[85] and genetic algorithms[86] in combination with MC simulations. This meant that by 1997, the first complete redesign of a naturally existing protein, the 28-residue the Zn(II) finger was achieved by Dahiyat and Mayo[87]. In 2003, the first novel protein fold, an alpha-beta protein called Top7, was designed by Brian Kuhlman and David Baker et al. who used fragment assembly (see also Chapter 3) to sample the novel backbone[88]. They utilized Rosetta, created in 1998, for this purpose and Rosetta arguably became the most influential software for protein design in the following two decades.

4.3.2 After Top7

After Top7, the protein design field expanded rapidly and here is an account up until the recent introduction of DL-based design methods. With increased computer power, lowering cost of gene synthesis, testing of design principles, perhaps combined with increased interest and funding, meant more advanced design could be pursued. In this period, particularly through Rosetta, we saw the development of many kinds of *de novo* proteins. Many different assemblies were designed, from fibers[89], 2D-arrays[90], cages (reviewed in detail later), 3D-crystals[91] to recent examples of mechanical rotors[92]. Enzymes[93] and ligand binding proteins[94], even ones that can imitate the functional properties of GFP[95]. Proteins with different folds from pure alpha[96], to alpha-beta[97] and purely beta[98]. Protein binders that can bind the stem of the influenza virus[99], interleukins involved in cancer[100] and more recently the corona spike protein[101]. Membrane proteins[102], hyper stable mini proteins[103], repeat proteins[104], and recently protein switches[105]. In these last two decades *de novo* protein design came of age and CPD has proven to be a versatile technique for designing novel proteins.

An attempt to enumerate all the great designs and breakthroughs made by a multitude of different scientists while keeping it brief (as promised in the title) will surely only do injustice. For excellent reviews on the *de novo* protein design field up until recent DL methods, I can refer the reader to reviews by Woolfson[106], Huang[73], Korendovych[107] and Pan[108].

4.3.3 Introduction of DL

Protein structure prediction and protein design have always been tightly interwoven fields, with many groups dipping their toes in both (ours included). With DeepMind's success in structure prediction in CASP14, the research into DL methods accelerated in the protein design field. After AlphaFold, many different DL-based methods for backbone and sequence design started to appear (see Ferruz et al.[109] for an excellent review) based on many different architectures such as CNNs, VAEs, GANs, GNNs and Transformers. Initially, however, experimental validation was lagging. Experimental validation has always been a cornerstone of the protein design field as it is the ultimate test of the method. Experimental testing allows us to assess the real-world behaviour of the designed protein, such as solubility and expression levels, and we can check if the experimental structure matches our design. Many DL-based sequence design algorithms, for instance, are often benchmarked on their native sequence recovery (NSR) capabilities. However, the ability to recapitulate native sequences, which is usually not better than ~50%, does not necessarily translate into real-world success when the proteins are made in the lab. Therefore, it has not been until recently that the new DL-based methods, which have been tested and validated in the lab, have started to be widely adopted.

Recently, the DL-based structure prediction network trRosetta was essentially trained to run in reverse to 'hallucinate' new proteins[110]. The method was experimentally tested to produce accurately predicted structures. Later, RoseTTAFold, trRosetta's predecessor, was similarly run in reverse to design proteins to scaffold functional sites using methods called constrained hallucination and inpainting with designs also being experimentally validated[111]. The most recent example is the development of RFdiffusion[112]. RFdiffusion is essentially a general design DL-software that can be used for a wide range of different protein targets. It uses a diffusion model, where RoseTTAFold is used to iteratively denoise a noisy signal to finally produce a protein. It can design proteins with *de novo* folds and has shown significant improvements to previous methods such as hallucination. While the authors of RFdiffusion, experimented with both backbone and sequence design, better sequence design performance was achieved through another DL-based method called ProteinMPNN[113]. Both RFdiffusion and ProteinMPNN have been experimentally validated and are the current state-of-the-art in backbone generation and sequence design, respectively.

4.4 Old and New

The last couple of years have been a transition period where DL-based methods have slowly started to take over the protein design methodology. However, the conventional strategy is still relevant and is being used alongside the current DL-based methods. Here I think it is useful to juxtapose the pure conventional design methodology with the newer pure DL-based methodology to highlight how they differ. The two methodologies are shown in **Figure 9** and are divided into three *in silico* stages of protein design: backbone design, sequence design and evaluation.

4.4.1 Conventional Design Strategy

In conventional protein design, a backbone is generated either using backbone templates from the PDB, or *de novo* by fragment assembly utilizing a blueprint[114]. Once a satisfactory backbone has been acquired, a sequence is designed onto the backbone using a score function that optimizes the combination of amino acids and rotamers given the backbone. The most widely used design algorithm before ProteinMPNN was the FastDesign[115] algorithm, implemented in Rosetta, which combines an iterative sequence search with a backbone search. Finally, the evaluation of the designed models involves a combination of structure prediction, filters, and human intuition. For smaller proteins, the structure of the designs can be predicted using *ab initio* predictions, with the goal of the predictions matching the designs. This is also called forward folding. Filters are also extensively applied depending on the design targets. In **Figure 9**, common filters for designing assemblies are highlighted, which include the bound-unbound energy difference (ΔG), change in Solvent Accessible Surface Area (ΔSASA) and Shape Complementarity (SC). At the very end, human intuition is applied by manually looking at the structures through graphical software, to assess which proteins should be tested experimentally.

4.4.2 DL Design Strategy

In comparison, DL methods generally start out with an NN that generates a backbone, with the current state-of-the-art being RFdiffusion. This is then followed by another NN that generates the sequence, with the current state-of-the-art being the message passing NN ProteinMPNN. Finally, AF/AFM is used to predict the structure and the design is compared to the prediction through RMSD, predicted LDDT (pLDDT) and predicted TM (pTM).

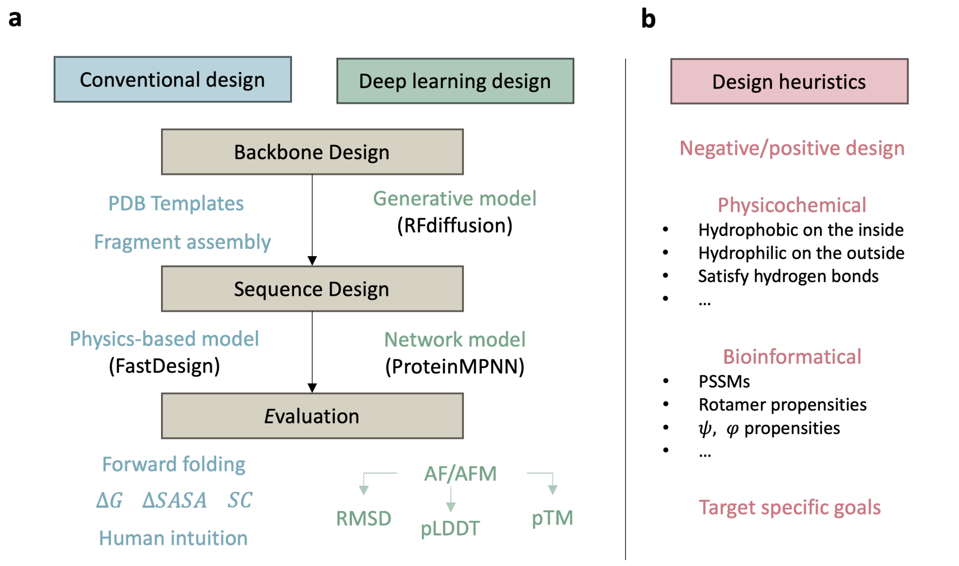

Figure 9: Design methodology for conventional and DL-based protein design.
a: Pure conventional design (blue) and pure DL-based design (green) methodologies juxtaposed. The three stages of protein design are shown in brown. In parenthesis, the state-of-the-art methods are highlighted. The components are discussed in the text. **b**: Examples of design heuristics. Negative design heuristics are measures that disfavour other states than the design target, while positive design heuristics are measures that favour the design state. Other design heuristics include physiochemical and bioinformatical measures and more specialized ones that relate to the target being designed. In the conventional design strategy, these are explicitly modelled, while in the DL methods, these are implicitly learned.

Protein design has always been a bit of an art form, where several design heuristics, gathered over decades of experimentation, had to be explicitly implemented in the design procedure (**Figure 9**). A good example of this is the Rosetta XML-scripts[116] which easily could be several pages long, specifying which heuristic should be applied where and when.

In contrast, the DL-based methods can learn at least some of these heuristics during training, and most likely, extract information beyond what we currently can understand.

4.5 Design of Protein Cages.

Now that we have reviewed the field at large it is time to be specific. In **paper III** we designed protein assemblies which are part of a larger class of proteins that are commonly called polyhedral assemblies or cages. Cages are roughly spherical protein assemblies with a hollow container and at least 2 unique interfaces[117, 118]. Natural examples include viral capsids (see also Chapter 1), ferritin[119], clathrin[120] and bacterial microcompartments[121]. There has been considerable interest in *de novo* designing protein cages as their architecture lends itself to many different applications. Their surface can be used as a display scaffold for applications such as antigen presentation for vaccines[122] and their interior used as a delivery vehicle in drug delivery[122] or as nanoreactors[123]. Here I will review the literature for the *de novo* design of protein cages.

I have divided cage design into 6 main strategies that are employed. Two based on a genetic strategy: Oligomer-fusion and Coiled coils mediation. Two based on designing the interface between building blocks: Metal coordination and Protein-Protein-Interface (PPI) design. One intermediary between the fusion and interface design: Rigid hierarchical fusion**.** The last of the 6 categories is the new versions of cage designs that are based on DL frameworks: DL-based design.

As I go along, I will discuss their advantages and limitations. At the end of the review, I will come back to how **Paper III** fits into the picture.

4.5.1 Oligomer-Fusion

The simplest strategy in cage design is to genetically fuse natural protein oligomers together through a semi-rigid linker such as an alpha-helix[124]. In this case, the natural protein oligomer constitutes one of the interfaces and the fusion interface the other, and the semi-rigid linker helps preserve the structural integrity to some degree. The main advantage of this strategy is its straightforwardness and simplicity. However, as rigidity is only partly preserved, one of the limitations of oligomeric fusion strategies is their structural polymorphism. This makes it difficult to control the final shape and can make the cage difficult to characterize structurally. Any small flexibility will propagate throughout the assembly making it difficult to design larger assemblies[125, 126, 127].

Recently, it was shown that by fusing 3 independent domains together, it was possible to create an icosahedral cage consisting of 60 subunits[128]. Nevertheless, it still showed substantial flexibility and asymmetric deformation. Another limitation is that the linkers used should end in a helical conformation and the connection point should be accessible for the oligomers to be connected.

4.5.2 Coiled Coils Mediation

Another strategy, akin to the oligomer-fusion strategy, is to use coiled coils, either using orthogonal pairs[129] or non-orthogonal pairs[130, 131]. This approach has been combined with disulphide bonds to create very large (100 nm) protein cages[132] similar to how DNA origami structures[133] are assembled to create cage structures.[134] Coiled coils are well-studied and their amenability for designability, orthogonality and varying oligomerization states make them attractive building blocks. Nevertheless, as a fusion strategy, they can suffer from similar flexibility problems as the oligomer-fusion strategies.

4.5.3 Metal Coordination

One non-genetic fusion strategy, but an interface design strategy, is to mediate the interfaces between protein scaffolds through metal sites.[135] Examples include coordination through Zn^{2+} and Cu^{2+} to create a tetrahedral cage[136], Zn^{2+} and Fe^{3+} to create octahedral cages[137] and Au^{2+} and Hg^{2+} to create even higher order symmetries[138]. Metal-coordinated cages are particularly interesting because the interface design only requires designing around the binding motif.

Metal coordination lends itself very well for switching mechanics as the metal can easily be removed via different mechanisms such as by chelation[136]. Nevertheless, metal coordination does have limitations. First, the protein backbone scaffolds need to have specific orientations to accommodate the metal ions. Second, there is only a limited amount of metal ions that can be used, limiting the total structural design space.

4.5.4 PPI Design

Protein-protein interface design relies on computationally designing the interface between existing oligomers. This has been a very successful strategy and many kinds of different symmetrical cages have been designed, from single component structures[139, 140] to two-component structures[141, 142], to more recent structures incorporating quasi and pseudo symmetry[143]. Two steps are employed: First, a symmetrical docking step followed by a design step. Docking strategies have evolved from TCDock[142], to sicdock[144], to recently RPXDock[145]. Sequence design has evolved from versions of FastDesign but is now switching to ProteinMPNN. PPI design has proven to be a very powerful approach, creating very stable assemblies, with applications that are now being explored in areas such as drug delivery[146] and vaccine design[147]. Traditionally, natural protein oligomers have been used to form one interface while the other has been designed. Multiple interfaces can be designed in a 2-step process by first designing 1 oligomer (with C5 symmetry for instance), followed by then designing it into a cage[143]. While PPI design is versatile, they are perhaps the most complicated to design with a success rate of about 10% per design[148].

4.5.5 Rigid Hierarchical Fusion

Rigid hierarchical fusion is an approach that rigidly fuses (HelixFuse) or docks (HelixDock) together a combination of designed helical repeat proteins (DHRs)[149] and helical bundles (HBs)[96, 150, 151, 152] using a computational search and sequence design strategy called WORMS[153]. These designs have been used to create cages with different symmetries[153, 154], and recently combined with antibodies to create functional cages[155]. These designs are generally more rigid than previous linker-based designs but also involves more computational design, which makes them less straightforward. As the linkages are designed and fused on the computer, they do offer flexibility in their designability of shape and structure, but as connections must be made by helical fusion it limits the total structural design space. Nonetheless, these systems are very modular, particularly with the new wave of designs that show very high modularity[156] and the possibility to incorporate pseudosymmetry[157].

4.5.6 DL-based Designs

Hallucination is currently not able to produce high-order symmetries[112, 158], but mainly two other strategies have been applied to cage design. The first is RFdiffusion, which can generate structures from scratch. The second is a top-down method that combines reinforcement-based learning with a Monte Carlo Tree Search (MCTS)[159]. Interestingly, in the MCTS strategy, different architecture constraints can be implemented in the loss function, such as shape and porosity of the capsid. The method can design small novel cages with very low porosity. While RFdiffusion and the MCTS method can design novel structures, the success rate is much lower than previously PPI designed structures. Another interesting diffusion-based model that is worth mentioning is Chroma[160], developed by Generate Biomedicines. It can generate very large, virus-like structures but lacks current experimental testing and has not yet been peer-reviewed.

Many different methodologies have been applied to design cage structures with varying levels of benefits and limitations. It is paramount that if we want to make cages that can be used for a myriad of different applications, we need to develop techniques that can better control the overall structure. A main feature of most of the cages designed to date, is that they are very porous, which doesn’t lend itself to applications, such as in drug delivery of small molecules, where the interior need to be sealed. To date, the MCTS has been the best strategy to address this problem, but the success rate is very low, which currently makes it not very generalizable.

One of the reasons for the higher success rate of the PPI design strategy is the use of pre-existing protein structures. Nevertheless, using pre-existing protein structures to design low porous cages has not been demonstrated, as it requires designing multiple interfaces and exquisite control of the protein shape. In **Paper III** we demonstrate how using existing natural protein scaffolds, combined with a shape-based protein design strategy, can yield structures with a wide range of shapes, sizes, and low porous assemblies.

5 Summary of Research Papers

Here I report the findings, and future directions of the **Papers I, II,** and **III.** Additional background for the papers can be found in Chapter 6. This includes the scoring metrics that are commonly used in docking benchmarks (such as in **Paper I**) and a mathematical description of the geometric shape matching in **Paper III**.

5.1 Paper I

In the last couple of years, we have seen a revolution in the protein structure prediction field with the introduction of the deep learning models AlphaFold (AF) and AlphaFold-Multimer (AFM). They both serve as state-of-the-art in monomeric and multimeric protein structure prediction respectively. Nonetheless, AFM shows declining performance the more chains that are modelled, and it has not been trained on structures above 9 chains and 1536 residues. Advances have been made to extend the performance of AFM by predicting subcomponents of larger assemblies. For instance, by predicting dimers and trimers, and then connecting them in various ways as discussed in Chapter 3. for MolPC and CombFold. This has meant that now up to 30 chains can be predicted in some cases. Nonetheless, prior to **Paper I,** the subcomponent assembly strategy had 2 major limitations. First, unrefined errors in the AFM predictions can lead to error propagation and mean the prediction of larger and larger assemblies can become difficult. Second, for protein complexes with multiple interfaces, one needs to rely on predicting more than 1 interface, which can be hard to extract from AlphaFold-Multimer.

In this study, we focus on some of the most complex protein assemblies in nature: cubic symmetrical assemblies (see Chapter 1.). We show how assemblies of up to 60 chains, including viral capsids, can be predicted with high accuracy. We predict monomeric and multimeric subcomponents from AF/AFM and arrange them and dock them symmetrically using SymEvoDOCK, a symmetrical extension to EvoDOCK (see Chapters 2 and 3) which features improvements in several areas for docking symmetrical protein assemblies.

Iteratively docking the assembly can reduce error propagation introduced by AFM, which combats the first problem of the subcomponent strategy. Additionally, utilizing symmetry for large assemblies is widely applicable for large protein assemblies as most protein complexes over 10 chains are symmetrical as discussed in **Paper I.**

We describe 3 areas of applications: local recapitulation, local assembly, and global assembly. Local recapitulation can be used to study energy landscapes by re-docking crystal structures or the like (as is done in **Paper II** for viral capsids). Local assembly can be used to refine protein structural models, for instance, Cryo-EM models, which have become a go-to method for large protein assemblies. Lastly, with global assembly, we show how large protein assemblies can be predicted *ab initio,* directly from sequence, without using structural templates. We also show that this can be accomplished by utilizing only 1 AFM prediction, combatting the second problem of the subcomponent strategy.

On a benchmark of 27 cubic symmetrical structures, we show that 22 out of 27 can be predicted below 2.0 Å RMSD with local recapitulation, with the remaining structures being between 3.6-9.4 Å RMSD. For local and global assembly, we cluster the results based on the 6 rigid body degrees of freedom into 5 clusters and output the best models in those clusters according to the interface score (Iscore). The Iscore is the Rosetta energy term between the bound and unbound state. We present both the best ranked model of the 5 according to the Iscore, and the best model of the 5 according to the metric we are looking at: TM-score, RMSD, and DockQ (see the additional background for how they are calculated). We write both the value of the metric as the best ranked model with regards to the Iscore, and the value of the best cluster model according to the metric separated by a dash ("/"). For local assembly, we find that we could predict structures with a median TM-score of 0.99/0.99, median DockQ score of 0.76/0.82, and median RMSD of 1.5/1.2 Å on the set of benchmark structures described previously. For global assembly, we find a median TM-Score of 0.99/0.99, DockQ score of 0.72/0.80, and RMSD of 1.6/1.5Å.

We also predict the subcomponents, consisting of the monomer, the 2-, 3- and 5-fold interfaces of 111 different cubic protein structures with AF/AFM. We find that AF/AFM can predict the subcomponents of cubic structures very well. In 78% of the cases, monomers can be predicted well (≤ 2 Å RMSD) to the native and in 72% of the cases, at least 1 interface can be predicted with AFM (≤ 2 Å RMSD).

Since our method is built on top of AF/AFM, we selected benchmark structures based on the ability of AF/AFM to predict them well, according to their quality metrics (pLDDT ≥ 90 and ipTM + pTM ≥ 0.9). However, not all sequences can be predicted with AF/AFM, but we find that 57% of cubic structures can be predicted to the level required for our benchmark set.

With everything taken together, we expect that 44%-50% of cubic structures can be predicted *ab initio* with the method presented here. However, while the benchmarked structures are themselves not in the AFM training set, they can share sequence homology with structures in the training set. However, the list of PDB depositions in the AFM training set has not been released so we cannot control for that. We calculated the sequence identity to all PDB files that could potentially be in the AFM training and focusing on the success rate for low homology structures ($\leq 30\%$) yields a success rate of 22% - but this is largely due to poorer AFM predictions.

5.2 Paper I: Future directions

The most obvious next steps would be to generalize this method to other symmetrical assemblies such as assemblies with cyclical, dihedral, helical and crystal symmetry. The assemblies predicted here are also the simplest kind of cubic assemblies as they consist of 1 single homomeric subunit. Larger cubic structures or other symmetrical structures can have multiple subunits and be heteromeric as well. It would be possible to extend the method to larger assemblies, either by introducing more degrees of freedom, or perhaps better by treating parts of the structure as a single component, and then dock them as described here. For instance, a heteromeric capsid consisting of 2 unique subunits can be predicted with AFM and treated as a single subunit during docking.

Some symmetrical systems, in particular larger capsids, also utilize pseudo- or quasi-symmetry (see Chapter 1.). The symmetry machinery described here is built on Rosetta. While the ability to model quasi/pseudo symmetry has not previously been implemented in

Rosetta, efforts are underway to design pseudo- and quasi-symmetrical structures by Rosetta community members[143].

One thing that is not mentioned in the paper is that in practical use cases, the stoichiometry and symmetry type need to be known beforehand, as these are fixed in the simulation. The user can, however attempt different symmetries and stoichiometries and pick the best model from that (for instance, through the Iscore). The stoichiometry can be determined by many different experimental techniques such as mass spectroscopy, analytical ultra-centrifugation, or size exclusion chromatography. The identification of stoichiometry and symmetry can be aided by sequence searches to homologous structures. I expect that the determination of symmetry or stoichiometry in advance is an area where DL-based methods can aid the method in the future.

The method described here could be benchmarked against existing tools such as CombFold and MolPC, which are described in Chapter 3. For instance, MolPC showed good performance on symmetrical structures. Symmetrical structures can be easier to predict than larger asymmetrical structures since they have, in general fewer unique interfaces.

Some modifications to the code would also be beneficial. For instance, while it is useful to be able to predict from single subcomponents alone, being able to additionally utilise multiple subcomponents will surely be beneficial. These could be used to start the search around those predicted interfaces and constrain and limit the search space. The 'termini-cutting' method described in Figure 3 in **Paper I** is a bit rudimentary and can be improved. It only considers the monomer, even in a complex. This is a disadvantage if the terminus forms significant interactions with its neighbours.

Developing a metric to pick out the best models based on more than just Iscore alone would also be beneficial. A confidence metric such as AF's pLDDT would also be desirable.

5.3 Paper II

Virus capsids exhibit some of the most complex self-assembly and disassembly mechanisms in nature. The assembly/disassembly mechanism is ultimately encoded in the protein interfaces, which for the simplest T1 virus capsid (see Chapter 1.) are the 5-, 3- and 2-fold interfaces. In this study, we carried out a comparison analysis of these interfaces in T1 capsids and compared them to regular homomeric structures with C_5, C_3 and C_2 symmetry. We wanted to see how they differ in order to get insight into the evolution of capsid structures and the assembly mechanisms. We analysed the binding energy, shape complementarity, interface size, number of contacts and amino acid compositions.

We found that viral interfaces are larger than their non-viral counterparts (especially the 3-fold). We also found differences in the amino acid compositions: capsid interfaces have more hydrophilic interfaces and are more proline enriched. Previous studies have suggested that larger interfaces in assemblies are more likely to be conserved[161], which in this case could indicate that the 3-fold interface is the driving block for evolution. The enrichment of proline could help the capsomers to conform to its many interfaces since it can drastically change the direction of the backbone.

We did not find any evidence for the quality of the interfaces being different in terms of binding energy or in their shape complementarity. This was surprising since experimental data and theoretical considerations[10, 162] have indicated that these interactions are weak. We hypothesise that the reason behind this could be due to two things unaccounted for in our free energy models: First, the long-range electrostatic forces, which we have used to model the binding energy. Second, the large conformational change and entropic cost of binding.

Furthermore, we analysed the energy landscapes by running SymEvoDOCK simulations (developed in **Paper I)** and found that capsids have deeply funnelled energy landscapes.

5.4 Paper II: Future directions

In general, some of the calculations could be improved. For the amino acid composition, we only considered amino acids with CB atoms in the interface, ignoring glycine. Glycine should be added to the discussion on polar/apolar interface differences, and it would be interesting to see if viral capsids are more glycine-rich than their non-viral counterparts. Glycine is the most flexible amino acid. If we saw an enrichment in capsid interfaces, it could be a contributing factor to the hypothesis raised that capsids use particular residues in the backbone, such as proline, to conform to the backbone.

The computational models used to model the binding energies could also be improved to give more insight into the two hypotheses raised about the electrostatic and entropic contributions of binding. Backbone relaxation can be added to the energy landscape analysis to assess what effect it would have on the assembly process.

5.5 Paper III:

Protein encapsulation systems, such as virus capsids, can be used for many applications (q.v. Chapter 4.). This includes uses as drug delivery systems or as scaffolds for powerful vaccines. Natural protein capsids can be used for these purposes, but evolution has optimized them for the fitness of the virus, and their usages are inherently limited. We wish to design them for new purposes with new architectures and functions. A fundamental challenge in protein design is to create encapsulation systems (also referred to as cages) with low porosity. This is a requirement for many applications, such as the delivery of small-molecule drugs. The problem with the design of cages is that you need to design multiple interfaces to achieve low porosity. This requires exquisite control over the shape of the subunit as it needs to fit into place with its many neighbours.

In **Paper III**, we describe a method to design cages with low porosity based on a principle we call shape-based protein design. The idea is based on geometric shape matching, for which the principle is first: to define what kinds of shapes can fit into a protein cage structure, and second: to find proteins with those shapes. The shape matching needs to be fast, and to this end, we use Zernike-Canterakis shape Descriptors (ZCD). In the additional background section in Chapter 6. this is described in further detail.

In the paper, we describe three approaches to designing protein cages based on shape. The first strategy is to use shapes of native capsids, as they often have shapes that are compatible with low porosities. The other two are based on 2D or 3D shapes, which can be simulated to achieve any characteristic desired, including low porosity.

In this paper, we explore the first strategy: creating new cage structures based on the shapes of native protein capsids. We designed 358 structures with a variety of shapes, sizes, porosities, and with different folds. We compared them to native capsids and found they have similar binding energies, shape complementarity and other structural properties. Furthermore, we discover that they have porosities similar to native capsids.

In the paper, we selected eight designs to analyse in detail based on their properties for being experimentally testable. One of the designs can be recapitulated in silico from AFM and forward docking using SymEvoDOCK. While it is not discussed in detail in the paper, these eight designs have been experimentally tested, and the results are under consideration (see next section).

5.6 Paper III: Future directions

The direction moving forward is two-fold. First, the computational protocol needs to be optimized. Second, we need to test more designs experimentally. When I left the project to focus on **Paper I** and **III**, the usage of DL-based tools for protein design and prediction of multimers was limited. Now, ProteinMPNN has become the state-of-the-art protein design method, and AFM has emerged as a useful tool to predict the structures of protein complexes. Implementing these two tools in the pipeline is going to be crucial. AF and AFM can be used for *in silico* screening. Only designs with a well-predicted monomer and at least one correct interface prediction should pass. One downside of AFM is that it can be time-consuming when screening many designs. As a first screen, ESMfold could perhaps be used. It is not quite as good as AFM, but it is very quick. A potential issue with AF/AFM in our pipeline is that we use building blocks from known protein structures, which could be available in the AF/AFM template library. Thus, AF/AFM could, to some degree, just report what it finds in the template library if the sequence has not been changed sufficiently.

How ProteinMPNN should be implemented in the protocol is under consideration. It can be used while docking before and/or after. The D1/D2 design strategy described in the paper is a little crude and will likely change completely in the future. ProteinMPNN is a fixed-backbone design algorithm and can be combined with a relax protocol, which has been reported elsewhere[163]. This might be crucial for our goal of designing cages as the backbone taken from existing structures (the PBBs described in the paper) needs to conform to the multiple interfaces. Another consideration is that, unfortunately, there's no good correlation between the shape descriptor used, the shape alignment score from ZEAL, and the metrics used for design validation (binding energy, etc.) for the designs, as shown in **Figure 10.** One caveat is that we did not restrict the docking space locally around the shape alignment, which could obscure the results here as the structures can move away from the shape they were aligned into.

This problem can also be intrinsic to ZCD itself. ZCD is just one of many shape descriptors, and we might consider using other ones in the future. Nonetheless, the resolution of ZCD can be increased in various ways. In the paper, we have used a low shape resolution, and increasing the resolution could potentially help to correlate the metrics better.

The other next challenge is experimental testing. The eight designs that are considered ready for experimental testing in the paper have been tested in the lab. The design called d2qeya2 was synthesized but was not very soluble. Two of the other designs show soluble expression, but the results need to be re-tested. ProteinMPNN has shown improved solubilization, which surely will be important in our case. Outside the designs discussed in the paper, we have tested the redesign of native capsids, where the interface was redesigned. 4/10 redesigns show soluble expression. Thus, we do have initial data that these could work in the lab and that our design methodology is sound.

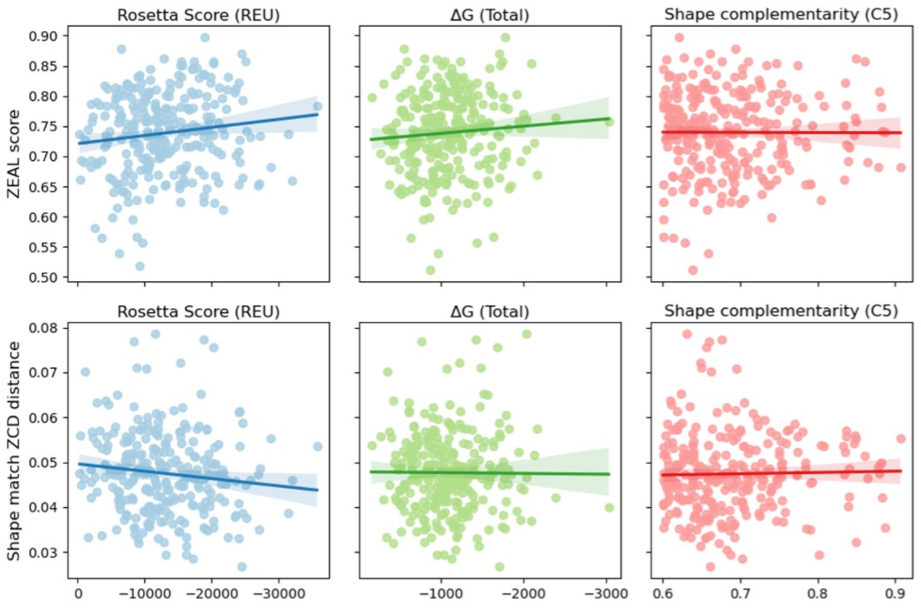

Figure 10: Zeal score and ZCD distance vs design quality metrics.
The top row shows the ZEAL score vs the Rosetta score, the total binding energy (ΔG), and shape complementarity (for the 5-fold interface only) for the D1 designs (see paper). The bottom row shows the shape match ZCD distance (see additional background) between the same metrics for D1 designs. The darker line shows the linear fit, and the shadow is the confidence interval.

6 Additional Background

6.1 Comparison metrics

When predicting the structures of proteins, it is important to evaluate how well the predicted models fit the experimentally validated structure. Usually, a form of distance metric or similarity metric is used. Here, I present 4 of such metrics.

6.1.1 RMSD

The most common distance-based metric is the Root Mean Square Deviation (RMSD). RMSD can be calculated over all atoms or a given subset of atoms, most commonly the Cα. The model and native coordinate atom pairs are superimposed on each other according to the atom selection of size n, and the RMSD is calculated as:

$$RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n} {d_i}^2} \qquad (6.1)$$

Where d_i is the distance between the atom pairs. In structure prediction, it is common to plot the energy against the RMSD of all predictions to confirm that the energy function correlates with native structure similarity and to check that the energy function can pick out the most similar models.

6.1.2 TM-score

RMSD is a very crude measure of similarity, as local dis-similarity can affect the global similarity [164]. This is often the case for the termini, which are often very flexible and, therefore do not have a fixed orientation. TM-score[165] (template modelling score) is a metric that attempts to alleviate this issue by calculating d_i as an inverse square. It uses a set of residues called a template to match the model onto the native structure. It is defined as:

$$TMscore = \text{Max}\left[\frac{1}{L_N}\sum_{i=1}^{L_T}\frac{1}{1+\left(\frac{d_i}{d_0}\right)^2}\right] \qquad (6.2)$$

Where L_N is the sequence length of the native structure and L_T the length of the aligned residues (the template) of the model. d_i is again the distance between the native and the aligned residues, and d_0 is a scaling factor that depends on L_N. “Max” refers to the procedure of identifying the best superposition of the native structure and the model. The TM-score ranges between (0, 1], where 1 is the best score. TM-align[166] – an alignment software based on the TM-score, is often used to find the best template to match the model onto the native.

6.1.3 lDDT

Instead of considering the global similarity, one can also consider local similarity. The local Distance Difference Test (lDDT)[166] is one such metric. It measures how well local atom pair distances are preserved in a model compared to the native structure. Each pairwise atom distance is calculated in the model within four different radius thresholds (called the inclusion radii). If the atom distances are preserved compared to the native structure within a distance threshold, the distance is set to be preserved else non-preserved. The average fraction of preserved to non-preserved atom pair distances for the four different radii thresholds is the final lDDT score.

6.1.4 DockQ

So far, we have discussed metrics for both monomeric and multimeric protein structures. The most widely accepted score for docked protein models is the DockQ score[167]. It considers the interaction between a dimer where the largest chain is referred to as the receptor and the smaller the ligand. It uses inverse square scaling, similar to the TM-score, of the RMSD, which is here called RMS (as in the original article) given by:

$$RMS_{scaled}(RMS, d_i) = \frac{1}{1 + \left(\frac{RMS}{d_i}\right)^2} \qquad (6.3)$$

Where d_i is a scaling parameter that determines how quickly the RMS approaches zero. The DockQ score is calculated as:

$$DockQ = \frac{F_{nat} + RMS_{scaled}(LRMS, d_1) + RMS_{scaled}(iRMS, d_2)}{3} \qquad (6.4)$$

F_{nat} is the fraction of native contacts that are preserved in the predicted interface defined by any heavy atoms within 5 Å. LRMS and iRMS are the RMSD of the ligand and of the interface, respectively, which in this case is defined by contacts within 10 Å.

6.2 Shape Matching

The idea behind shape matching is that a shape, such as the molecular surface of a protein, can be represented with a simple shape descriptor format, which can be quickly compared to other shapes. In **Paper III**, we use Zernike-Canterakis (ZC) shape descriptors[168], which have many desirable features for protein shape matching[169]:

Descriptive power: The descriptor can describe the shape well.

Compactness: It is easy to store computationally and quick to compare.

Rotational invariance: It does not depend on the orientation of the object.

The idea is to project a shape function: f(r), into a linear combination of ZC polynomials $Z_{nl}^{m}(r)$ weighted by ZC moments Ω_{nl}^{m} which can be written as:

$$f(r) = \sum_{n}^{N} \sum_{l}^{n} \sum_{m=-l}^{l} \Omega_{nl}^{m} Z_{nl}^{m}(r) \qquad (6.5)$$

The ZC polynomials ($Z_{nl}^{m}(r)$) are themselves functions that represent a shape and are often called 3D Zernike functions. The above equation approximates the real shape, and the higher the value of N, the better the shape is described.

Since the ZC moments Ω_{nl}^{m} are the weights, the values of those are what we can use to describe and compare shapes. However, the ZC moments are not rotationally invariant, so we need another step to make them so. The ZC moments are collected into 2l + 1-dimensional vectors:

$$\Omega_{nl} = \begin{vmatrix} \Omega_{nl}^{l} \\ \Omega_{nl}^{l-1} \\ \Omega_{nl}^{l-2} \\ \vdots \\ \Omega_{nl}^{-1} \end{vmatrix} \tag{6.6}$$

By taking the norm of them, we can define a rotationally invariant feature vector F_{nl}:

$$F_{nl} = \left\| \begin{matrix} \Omega_{nl}^{l} \\ \Omega_{nl}^{l-1} \\ \Omega_{nl}^{l-2} \\ \vdots \\ \Omega_{nl}^{-1} \end{matrix} \right\| \tag{6.7}$$

These feature vectors: F_{nl} , are collected into our final ZC shape descriptor (ZCD) vector:

$$\mathrm{ZCD} = [F_{00}, F_{20}, F_{22}, F_{31}, F_{33}, \dots] \tag{6.8}$$

The number of elements in a ZCD vector is described by its order: n, which is given by (n+1)/(n+4)/4 for odd *n* and (n/2+1)2 for even *n*. In **Paper III,** we use *n* = 20, an even number, which results in 121 elements. The more alike the numbers in the ZCD vector of two shapes are, the more alike these shapes are. The simplest comparison metric of two ZCD vectors is the Euclidian distance, which is also the distance metric used in **Paper III**.

7 Conclusion

In **Paper I,** we developed a method to predict large cubic symmetrical protein structures such as virus capsids from sequence. The method highlights how combining classical methods with deep learning approaches can be synergistic. Furthermore, the method described here can readily be extended to other symmetries and larger structures. New viruses emerge every year and give rise to a large number of human diseases. Being able to quickly predict their structure can help us to rapidly develop drugs to respond to them.

It is also important to understand the assembly mechanisms, which was done in **Paper II**. While the paper might have raised more questions than it answered, it gives an insight into the energy landscapes and differences between their interfaces compared to regular homomers.

While most people are horrified by viruses, we must not forget they are marvellous feats of natural engineering. They are small nanocontainers which can be used to deliver drugs or as vaccine scaffolds. While nature has evolved them for a limited range of uses, we wish to design them for new purposes with different architectures and functionality. In **Paper III,** we did just that. By explicitly modelling the shapes of the proteins in the capsid, we were able to create a range of different structures with different shapes, porosities, and sizes. These have characteristics similar to native capsids. We selected eight designs from our design pipeline, which are currently being experimentally tested in our lab.

While **Paper I** is a finished project that is currently in review, **Paper II** and, in particular, **Paper III** are still ongoing projects**.** While **Paper II** needs some finer adjustment before submission, **Paper III** need further work. As mentioned in Chapter 4., computational protein design is crucially linked to experimental testing. As described in Chapter 5. on future work for **Paper III,** we still want to improve upon the computational pipeline and incorporate many of the new DL-based tools into the pipeline. Also, more designs need to be tested.

8 Outlook

In this thesis, we have explored different fields. In **Paper I,** we developed tools for predicting protein structures. The protein structure prediction field is moving incredibly fast. It is even too fast for peer review, and to be at the forefront, you must be able to discern the validity of preprints on bioRvix. For instance, AlphaFold-Multimer has, as of writing, not been peer-reviewed but is currently in routine use by many research groups around the world.

The field has also entered a new era. Big Tech companies such as Google (AlphaFold) and Meta/Facebook (ESMfold) have entered the arena and are pushing it to new horizons. These companies have the capacity to put many top scientists and engineers together for a single purpose. By contrast, scientists in academia often work alone or in smaller groups, and this might be one of the reasons academia is lagging behind in the field. To truly accomplish big goals and push the frontier of science, I think it is necessary to work together as teams.

While it is great that Big Tech companies are pushing the field forward, the flip side of the coin is that the science can be hidden behind closed doors. If we are not able to use it, scrutinize it, and repeat it, is it really science? So far, there have been some hiccups. For instance, AlphaFold 1 could not be used to predict proteins beyond CASP13, and recently, AlphaMissense[170] has been released without the trained weights, which means no new predictions can be made. While I have no doubts that great and passionate scientists are working behind the scenes, the nature of companies is to make money, which, unfortunately, can hurt science. We will see what the future holds. Luckily, much has been made available to the public and implementations are still released to a large degree.

Another field we explored was protein design (qv. **Paper III**). While most of the state-of-the-art tools are now DL-based (RFdiffusion, ProteinMPNN), the field hasn't seen its big quantum leap, as we saw for structure prediction. While the new DL models are amazing achievements, most of the proteins tested in the lab still fail to express, solubilize, or fold. If you take another look, you will also notice that most of the designs are solid structures (jokingly often referred to as 'rocks'). Proteins are dynamic and functional molecular machines, and ultimately, this is what we want to engineer as protein designers. Nevertheless, more and more proteins are being designed with functions by amazing scientists around the world. A recent trip to a Rosetta conference – attended by a large community of protein designers, revealed that functions and conformational changes are the next big frontier researchers are working on.

In all the excitement about DL, we might forget that there are fundamental scientific questions it does not directly address. While DL models have become great at predicting protein structures, we should not forget the grand-old challenge of protein folding - which was not one-fold but multi-fold (pun intended). Predicting the structure was just one of the challenges raised; the 'how' questions, such as the mechanisms of protein folding, are still not fully understood[171]. Similarly, one of the main goals of doing protein design has been to test our understanding of proteins, as often championed through Richard Feynman's famous quote: "What I cannot create, I do not understand". However, current DL methods do not tell us about the physical principles being applied, which leaves us nowhere closer to the goal of truly understanding proteins.

9 References

1. Lopez MJ, Mohiuddin SS. Biochemistry, Essential Amino Acids. In: *StatPearls*) (2023).

2. Berman HM*, et al.* The Protein Data Bank. *Nucleic Acids Res* **28**, 235-242 (2000).

3. Sillitoe I*, et al.* CATH: increased structural coverage of functional space. *Nucleic Acids Res* **49**, D266-D273 (2021).

4. Fox NK, Brenner SE, Chandonia JM. SCOPe: Structural Classification of Proteins-extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res* **42**, D304-D309 (2014).

5. Ferruz N*, et al.* Identification and Analysis of Natural Building Blocks for Evolution-Guided Fragment-Based Protein Design. *J Mol Biol* **432**, 3898-3914 (2020).

6. Bergendahl LT, Marsh JA. Functional determinants of protein assembly into homomeric complexes. *Sci Rep-Uk* **7**, (2017).

7. Goodsell DS, Olson AJ. Structural symmetry and protein function. *Annu Rev Bioph Biom* **29**, 105-153 (2000).

8. Sudarev VV*, et al.* Ferritin self-assembly, structure, function, and biotechnological applications. *Int J Biol Macromol* **224**, 319-343 (2023).

9. Organization WH. COVID-19 Dashboard.). World Health Organization.

10. Perlmutter JD, Hagan MF. Mechanisms of Virus Assembly. *Annu Rev Phys Chem* **66**, 217-239 (2015).

11. Johnson JE, Speir JA. Quasi-equivalent viruses: A paradigm for protein assemblies. *J Mol Biol* **269**, 665-675 (1997).

12. Caspar DL, Klug A. Physical principles in the construction of regular viruses. *Cold Spring Harb Symp Quant Biol* **27**, 1-24 (1962).

13. Klose T, Kuznetsov YG, Xiao CA, Sun SY, McPherson A, Rossmann MG. The Three-Dimensional Structure of Mimivirus. *Intervirology* **53**, 268-273 (2010).

14. Leman JK*, et al.* Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nat Methods* **17**, 665-680 (2020).

15. Jorge N, Stephen JW. *Numerical optimization*. Spinger (2006).

16. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of State Calculations by Fast Computing Machines. *J Chem Phys* **21**, 1087-1092 (1953).

17. Chib S, Greenberg E. Understanding the Metropolis-Hastings Algorithm. *Am Stat* **49**, 327-335 (1995).

18. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. *Science* **220**, 671-680 (1983).

19. DiMaio F, Leaver-Fay A, Bradley P, Baker D, André I. Modeling Symmetric Macromolecular Structures in Rosetta3. *Plos One* **6**, (2011).

20. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* **521**, 436-444 (2015).

21. OpenAI T. Chatgpt: Optimizing language models for dialogue. OpenAI.) (2022).

22. Silver D*, et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484-+ (2016).

23. Norn C, André I. Atomistic simulation of protein evolution reveals sequence covariation and time-dependent fluctuations of site-specific substitution rates. *Plos Computational Biology* **19**, (2023).

24. Dill KA, MacCallum JL. The Protein-Folding Problem, 50 Years On. *Science* **338**, 1042-1046 (2012).

25. Hart WE, Istrail S. Robust proofs of NP-hardness for protein folding: General lattices and energy potentials. *J Comput Biol* **4**, 1-22 (1997).

26. Pearce R, Zhang Y. Toward the solution of the protein structure prediction problem. *J Biol Chem* **297**, (2021).

27. Jumper J*, et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583-+ (2021).

28. Evans R*, et al.* Protein complex prediction with AlphaFold-Multimer. *bioRxiv*, 2021.2010.2004.463034 (2022).

29. Dorn M, Silva MBE, Buriol LS, Lamb LC. Three-dimensional protein structure prediction: Methods and computational strategies. *Comput Biol Chem* **53**, 251-276 (2014).

30. Lindorff-Larsen K, Piana S, Dror RO, Shaw DE. How Fast-Folding Proteins Fold. *Science* **334**, 517-520 (2011).

31. Gershenson A, Gosavi S, Faccioli P, Wintrode PL. Successes and challenges in simulating the folding of large proteins. *J Biol Chem* **295**, 15-33 (2020).

32. Simons KT, Kooperberg C, Huang E, Baker D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J Mol Biol* **268**, 209-225 (1997).

33. Gobel U, Sander C, Schneider R, Valencia A. Correlated Mutations and Residue Contacts in Proteins. *Proteins* **18**, 309-317 (1994).

34. Xiang ZX. Advances in homology protein structure modeling. *Curr Protein Pept Sc* **7**, 217-227 (2006).

35. Bowie JU, Luthy R, Eisenberg D. A Method to Identify Protein Sequences That Fold into a Known 3-Dimensional Structure. *Science* **253**, 164-170 (1991).

36. Wang ZX. A re-estimation for the total numbers of protein folds and superfamilies. *Protein Eng* **11**, 621-626 (1998).

37. Finkelstein AV, Ptitsyn OB. Why Do Globular-Proteins Fit the Limited Set of Folding Patterns. *Prog Biophys Mol Bio* **50**, 171-190 (1987).

38. Bertoline LMF, Lima AN, Krieger JE, Teixeira SK. Before and after AlphaFold2: An overview of protein structure prediction. *Front Bioinform* **3**, 1120370 (2023).

39. Katchalskikatzir E, Shariv I, Eisenstein M, Friesem AA, Aflalo C, Vakser IA. Molecular-Surface Recognition - Determination of Geometric Fit between Proteins and Their Ligands by Correlation Techniques. *P Natl Acad Sci USA* **89**, 2195-2199 (1992).

40. Wodak SJ, Vajda S, Lensink MF, Kozakov D, Bates PA. Critical Assessment of Methods for Predicting the 3D Structure of Proteins and Protein Complexes. *Annu Rev Biophys* **52**, 183-206 (2023).

41. Kozakov D*, et al.* The ClusPro web server for protein-protein docking. *Nat Protoc* **12**, 255-278 (2017).

42. Pierce BG, Wiehe K, Hwang H, Kim BH, Vreven T, Weng ZP. ZDOCK server: interactive docking prediction of protein-protein complexes and symmetric multimers. *Bioinformatics* **30**, 1771-1773 (2014).

43. Harmalkar A, Gray JJ. Advances to tackle backbone flexibility in protein docking. *Curr Opin Struc Biol* **67**, 178-186 (2021).

44. Desta IT, Porter KA, Xia B, Kozakov D, Vajda S. Performance and Its Limits in Rigid Body Protein-Protein Docking. *Structure* **28**, 1071-+ (2020).

45. Brooks BR, Bruccoleri RE, Olafson BD, States DJ, Swaminathan S, Karplus M. Charmm - a Program for Macromolecular Energy, Minimization, and Dynamics Calculations. *J Comput Chem* **4**, 187-217 (1983).

46. Cao Y, Musah RA, Wilcox SK, Goodin DB, McRee DE. Protein conformer selection by ligand binding observed with crystallography. *Protein Sci* **7**, 72-78 (1998).

47. Burman SSR, Yovanno RA, Gray JJ. Flexible Backbone Assembly and Refinement of Symmetrical Homomeric Complexes. *Structure* **27**, 1041-+ (2019).

48. Marze NA, Burman SSR, Sheffler W, Gray JJ. Efficient flexible backbone protein-protein docking for challenging targets. *Bioinformatics* **34**, 3461-3469 (2018).

49. Varela D, Karlin V, Andre I. A memetic algorithm enables efficient local and global all-atom protein-protein docking with backbone and side-chain flexibility. *Structure* **30**, 1550-+ (2022).

50. AlQuraishi M. AlphaFold at CASP13. *Bioinformatics* **35**, 4862-4865 (2019).

51. Porta-Pardo E, Ruiz-Serra V, Valentini S, Valencia A. The structural coverage of the human proteome before and after AlphaFold. *PLoS Comput Biol* **18**, e1009818 (2022).

52. Tunyasuvunakool K*, et al.* Highly accurate protein structure prediction for the human proteome. *Nature* **596**, 590-+ (2021).

53. Terwilliger TC*, et al.* AlphaFold predictions are valuable hypotheses, and accelerate but do not replace experimental structure determination. *bioRxiv*, 2022.2011.2021.517405 (2023).

54. Ruff KM, Pappu RV. AlphaFold and Implications for Intrinsically Disordered Proteins. *J Mol Biol* **433**, (2021).

55. Azzaz F, Yahi N, Chahinian H, Fantini J. The Epigenetic Dimension of Protein Structure Is an Intrinsic Weakness of the AlphaFold Program. *Biomolecules* **12**, (2022).

56. Perrakis A, Sixma TK. AI revolutions in biology The joys and perils of AlphaFold. *Embo Rep* **22**, (2021).

57. Saldano T*, et al.* Impact of protein conformational diversity on AlphaFold predictions. *Bioinformatics* **38**, 2742-2748 (2022).

58. Buel GR, Walters KJ. Can AlphaFold2 predict the impact of missense mutations on structure? *Nat Struct Mol Biol* **29**, 1-2 (2022).

59. Terwilliger TC*, et al.* Improved AlphaFold modeling with implicit experimental information. *Nat Methods* **19**, 1376-+ (2022).

60. Hekkelman ML, de Vries I, Joosten RP, Perrakis A. AlphaFill: enriching AlphaFold models with ligands and cofactors. *Nat Methods* **20**, 205-+ (2023).

61. Sala D, Engelberger F, Mchaourab HS, Meiler J. Modeling conformational states of proteins with AlphaFold. *Curr Opin Struc Biol* **81**, (2023).

62. Mirdita M, Schutze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M. ColabFold: making protein folding accessible to all. *Nat Methods* **19**, 679-+ (2022).

63. Elofsson A. Progress at protein structure prediction, as seen in CASP15. *Curr Opin Struc Biol* **80**, (2023).

64. Baek M, Anishchenko I, Humphreys IR, Cong Q, Baker D, DiMaio F. Efficient and accurate prediction of protein structure using RoseTTAFold2. *bioRxiv*, 2023.2005.2024.542179 (2023).

65. Lin ZM*, et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123-1130 (2023).

66. Marsh JA, Teichmann SA. Structure, Dynamics, Assembly, and Evolution of Protein Complexes. *Annu Rev Biochem* **84**, 551-575 (2015).

67. Bryant P, Pozzati G, Zhu WS, Shenoy A, Kundrotas P, Elofsson A. Predicting the structure of large protein complexes using AlphaFold and Monte Carlo tree search. *Nat Commun* **13**, (2022).

68. Shor B, Schneidman-Duhovny D. Predicting structures of large protein assemblies using combinatorial assembly algorithm and AlphaFold2. *bioRxiv*, 2023.2005.2016.541003 (2023).

69. Ahdritz G*, et al.* OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *bioRxiv*, 2022.2011.2020.517210 (2023).

70. Li Z*, et al.* Uni-Fold Symmetry: Harnessing Symmetry in Folding Large Protein Complexes. *bioRxiv*, 2022.2008.2030.505833 (2022).

71. Taylor WR, Chelliah V, Hollup SM, MacDonald JT, Jonassen I. Probing the "Dark Matter" of Protein Fold Space. *Structure* **17**, 1244-1252 (2009).

72. Chiarabelli C, De Lucrezia D, Stano P, Luisi PL. The World of the "Never Born Proteins". *Origins Life Evol B* **39**, 308-309 (2009).

73. Huang PS, Boyken SE, Baker D. The coming of age of de novo protein design. *Nature* **537**, 320-327 (2016).

74. Ade PAR*, et al.* Planck 2015 results XIII. Cosmological parameters. *Astron Astrophys* **594**, (2016).

75. Packer MS, Liu DR. Methods for the directed evolution of proteins. *Nat Rev Genet* **16**, 379-394 (2015).

76. Fisher MA, McKinley KL, Bradley LH, Viola SR, Hecht MH. De Novo Designed Proteins from a Library of Artificial Sequences Function in Escherichia Coli and Enable Cell Growth. *Plos One* **6**, (2011).

77. Keefe AD, Szostak JW. Functional proteins from a random-sequence library. *Nature* **410**, 715-718 (2001).

78. Kendrew JC, Bodo G, Dintzis HM, Parrish RG, Wyckoff H, Phillips DC. 3-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. *Nature* **181**, 662-666 (1958).

79. Bernstein FC*, et al.* Protein Data Bank - Computer-Based Archival File for Macromolecular Structures. *J Mol Biol* **112**, 535-542 (1977).

80. Regan L, Degrado WF. Characterization of a Helical Protein Designed from 1st Principles. *Science* **241**, 976-978 (1988).

81. Degrado WF, Regan L, Ho SP. The Design of a 4-Helix Bundle Protein. *Cold Spring Harb Sym* **52**, 521-526 (1987).

82. Ho SP, Eisenberg D, Degrado WF. The Design and Synthesis of a New 4-Alpha Helical Bundle Protein. *Abstr Pap Am Chem S* **194**, 197-Orgn (1987).

83. Degrado WF, Ho SP, Weber PC, Wasserman ZR, Eisenberg D. The Design, Synthesis, and Characterization of a 4-Helical Bundle Protein. *J Cell Biochem*, 196-196 (1987).

84. Woolfson DN. Understanding a protein fold: The physics, chemistry, and biology of alpha-helical coiled coils. *J Biol Chem* **299**, (2023).

85. Desmet J, Demaeyer M, Hazes B, Lasters I. The Dead-End Elimination Theorem and Its Use in Protein Side-Chain Positioning. *Nature* **356**, 539-542 (1992).

86. Jones DT. De-Novo Protein Design Using Pairwise Potentials and a Genetic Algorithm. *Protein Sci* **3**, 567-574 (1994).

87. Dahiyat BI, Mayo SL. De novo protein design: Fully automated sequence selection. *Science* **278**, 82-87 (1997).

88. Kuhlman B, Dantas G, Ireton GC, Varani G, Stoddard BL, Baker D. Design of a novel globular protein fold with atomic-level accuracy. *Science* **302**, 1364-1368 (2003).

89. Shen H*, et al.* De novo design of self-assembling helical protein filaments. *Science* **362**, 705-+ (2018).

90. Gonen S, DiMaio F, Gonen T, Baker D. Design of ordered two-dimensional arrays mediated by noncovalent protein-protein interfaces. *Science* **348**, 1365-1368 (2015).

91. Li Z*, et al.* Accurate Computational Design of 3D Protein Crystals. *bioRxiv*, 2022.2011.2018.517014 (2022).

92. Courbet A*, et al.* Computational design of mechanically coupled axle-rotor protein assemblies. *Science* **376**, 383-+ (2022).

93. Jiang L*, et al.* De novo computational design of retro-aldol enzymes. *Science* **319**, 1387-1391 (2008).

94. Tinberg CE*, et al.* Computational design of ligand-binding proteins with high affinity and selectivity. *Nature* **501**, 212-+ (2013).

95. Dou JY*, et al.* De novo design of a fluorescence-activating beta-barrel. *Nature* **561**, 485-+ (2018).

96. Huang PS*, et al.* High thermodynamic stability of parametrically designed helical bundles. *Science* **346**, 481-485 (2014).

97. Koga N*, et al.* Principles for designing ideal protein structures. *Nature* **491**, 222-227 (2012).

98. Marcos E*, et al.* De novo design of a non-local beta-sheet protein with high stability and accuracy. *Nat Struct Mol Biol* **25**, 1028-+ (2018).

99. Fleishman SJ*, et al.* Computational Design of Proteins Targeting the Conserved Stem Region of Influenza Hemagglutinin. *Science* **332**, 816-821 (2011).

100. Silva DA*, et al.* De novo design of potent and selective mimics of IL-2 and IL-15. *Nature* **565**, 186-+ (2019).

101. Cao LX*, et al.* De novo design of picomolar SARS-CoV-2 miniprotein inhibitors. *Science* **370**, 426-+ (2020).

102. Xu CF*, et al.* Computational design of transmembrane pores. *Nature* **585**, 129-+ (2020).

103. Bhardwaj G*, et al.* Accurate de novo design of hyperstable constrained peptides. *Nature* **538**, 329-+ (2016).

104. Parmeggiani F, Huang PS. Designing repeat proteins: a modular approach to protein design. *Curr Opin Struc Biol* **45**, 116-123 (2017).

105. Langan R*, et al.* De Novo Design of Bioactive Protein Switches. *Protein Sci* **28**, 47-48 (2019).

106. Woolfson DN. A Brief History of De Novo Protein Design: Minimal, Rational, and Computational. *J Mol Biol* **433**, (2021).

107. Korendovych IV, DeGrado WF. De novo protein design, a retrospective. *Q Rev Biophys* **53**, (2020).

108. Pan XJ, Kortemme T. Recent advances in de novo protein design: Principles, methods, and applications. *J Biol Chem* **296**, (2021).

109. Ferruz N, Heinzinger M, Akdel M, Goncearenco A, Naef L, Dallago C. From sequence to function through structure: Deep learning for protein design. *Comput Struct Biotec* **21**, 238-250 (2023).

110. Anishchenko I*, et al.* De novo protein design by deep network hallucination. *Nature* **600**, 547-+ (2021).

111. Wang J*, et al.* Scaffolding protein functional sites using deep learning. *Science* **377**, 387-+ (2022).

112. Watson JL*, et al.* De novo design of protein structure and function with RFdiffusion. *Nature* **620**, 1089-1100 (2023).

113. Dauparas J*, et al.* Robust deep learning-based protein sequence design using ProteinMPNN. *Science* **378**, 49-55 (2022).

114. Huang PS*, et al.* RosettaRemodel: A Generalized Framework for Flexible Backbone Protein Design. *Plos One* **6**, (2011).

115. Maguire JB*, et al.* Perturbing the energy landscape for improved packing during computational protein design. *Proteins-Structure Function and Bioinformatics* **89**, 436-449 (2021).

116. Fleishman SJ*, et al.* RosettaScripts: A Scripting Language Interface to the Rosetta Macromolecular Modeling Suite. *Plos One* **6**, (2011).

117. Lai YT, King NP, Yeates TO. Principles for designing ordered protein assemblies. *Trends Cell Biol* **22**, 653-661 (2012).

118. Norn CH, Andre I. Computational design of protein self-assembly. *Curr Opin Struc Biol* **39**, 39-45 (2016).

119. Liu XF, Theil EC. Ferritins: Dynamic management of biological iron and oxygen chemistry. *Accounts Chem Res* **38**, 167-175 (2005).

120. Royle SJ. The cellular functions of clathrin. *Cell Mol Life Sci* **63**, 1823-1832 (2006).

121. Rae BD, Long BM, Badger MR, Price GD. Functions, Compositions, and Evolution of the Two Types of Carboxysomes: Polyhedral Microcompartments That Facilitate CO2 Fixation in Cyanobacteria and Some Proteobacteria. *Microbiol Mol Biol R* **77**, 357-379 (2013).

122. Shirbaghaee Z, Bolhassani A. Different Applications of Virus-Like Particles in Biology and Medicine: Vaccination and Delivery Systems. *Biopolymers* **105**, 113-132 (2016).

123. Maity B, Fujita K, Ueno T. Use of the confined spaces of apo-ferritin and virus capsids as nanoreactors for catalytic reactions. *Curr Opin Chem Biol* **25**, 88-97 (2015).

124. Padilla JE, Colovos C, Yeates TO. Nanohedra: Using symmetry to design self assembling protein cages, layers, crystals, and filaments. *P Natl Acad Sci USA* **98**, 2217-2221 (2001).

125. Lai YT, Jiang L, Chen WY, Yeates TO. On the predictability of the orientation of protein domains joined by a spanning alpha-helical linker. *Protein Eng Des Sel* **28**, 491-499 (2015).

126. Lai YT*, et al.* Structure of a designed protein cage that self-assembles into a highly porous cube. *Nat Chem* **6**, 1065-1071 (2014).

127. Lai YT, Cascio D, Yeates TO. Structure of a 16-nm Cage Designed by Using Protein Oligomers. *Science* **336**, 1129-1129 (2012).

128. Cannon KA, Nguyen VN, Morgan C, Yeates TO. Design and Characterization of an Icosahedral Protein Cage Formed by a Double-Fusion Protein Containing Three Distinct Symmetry Elements. *Acs Synth Biol* **9**, 517-524 (2020).

129. Indelicato G*, et al.* Principles Governing the Self-Assembly of Coiled-Coil Protein Nanoparticles. *Biophys J* **110**, 646-660 (2016).

130. Patterson DP, Su M, Franzmann TM, Sciore A, Skiniotis G, Marsh ENG. Characterization of a highly flexible self-assembling protein system designed to form nanocages. *Protein Sci* **23**, 190-199 (2014).

131. Cristie-Dayid AS*, et al.* Coiled-Coil-Mediated Assembly of an Icosahedral Protein Cage with Extremely High Thermal and Chemical Stability. *J Am Chem Soc* **141**, 9207-9216 (2019).

132. Fletcher JM*, et al.* Self-Assembling Cages from Coiled-Coil Peptide Modules. *Science* **340**, 595-599 (2013).

133. Dey S*, et al.* DNA origami. *Nat Rev Method Prime* **1**, (2021).

134. Lapenta F*, et al.* Self-assembly and regulation of protein cages from pre-organised coiled-coil modules. *Nat Commun* **12**, (2021).

135. Sontz PA, Song WJ, Tezcan FA. Interfacial metal coordination in engineered protein and peptide assemblies. *Curr Opin Chem Biol* **19**, 42-49 (2014).

136. Cristie-David AS, Marsh ENG. Metal-dependent assembly of a protein nano-cage. *Protein Sci* **28**, 1620-1629 (2019).

137. Golub E*, et al.* Constructing protein polyhedra via orthogonal chemical interactions. *Nature* **578**, 172-+ (2020).

138. Malay AD*, et al.* An ultra-stable gold-coordinated protein cage displaying reversible assembly. *Nature* **569**, 438-+ (2019).

139. Hsia Y*, et al.* Design of a hyperstable 60-subunit protein icosahedron (vol 535, pg 136, 2016). *Nature* **540**, (2016).

140. King NP*, et al.* Computational Design of Self-Assembling Protein Nanomaterials with Atomic Level Accuracy. *Science* **336**, 1171-1174 (2012).

141. Bale JB*, et al.* Accurate design of megadalton-scale two-component icosahedral protein complexes. *Science* **353**, 389-394 (2016).

142. King NP*, et al.* Accurate design of co-assembling multi-component protein nanomaterials. *Nature* **510**, 103-+ (2014).

143. Dowling QM*, et al.* Hierarchical design of pseudosymmetric protein nanoparticles. *bioRxiv*, 2023.2006.2016.545393 (2023).

144. Fallas JA*, et al.* Computational design of self-assembling cyclic protein homo-oligomers. *Nat Chem* **9**, 353-360 (2017).

145. Sheffler W*, et al.* Fast and versatile sequence-independent protein docking for nanomaterials design using RPXDock. *Plos Computational Biology* **19**, (2023).

146. Edwardson TGW, Mori T, Hilvert D. Rational Engineering of a Designed Protein Cage for siRNA Delivery. *J Am Chem Soc* **140**, 10439-10442 (2018).

147. Walls AC*, et al.* Elicitation of Potent Neutralizing Antibody Responses by Designed Protein Nanoparticle Vaccines for SARS-CoV-2. *Cell* **183**, 1367-+ (2020).

148. Mallik BB, Stanislaw J, Alawathurage TM, Khmelinskaia A. De Novo Design of Polyhedral Protein Assemblies: Before and After the AI Revolution. *Chembiochem*, (2023).

149. Brunette TJ*, et al.* Exploring the repeat protein universe through computational protein design. *Nature* **528**, 580-+ (2015).

150. Boyken SE. De novo design of protein homo-oligomers with modular hydrogen-bond network-mediated specificity (vol 352, aag1318, 2016). *Science* **353**, 879-879 (2016).

151. Chen ZB*, et al.* Programmable design of orthogonal protein heterodimers. *Nature* **565**, 106-+ (2019).

152. Thomson AR*, et al.* Computational design of water-soluble alpha-helical barrels. *Science* **346**, 485-488 (2014).

153. Hsia Y*, et al.* Design of multi-scale protein complexes by hierarchical building block fusion. *Nat Commun* **12**, (2021).

154. Vulovic I*, et al.* Generation of ordered protein assemblies using rigid three-body fusion. *P Natl Acad Sci USA* **118**, (2021).

155. Divine R*, et al.* Designed proteins assemble antibodies into modular nanocages. *Science* **372**, 47-+ (2021).

156. Huddy TF*, et al.* Blueprinting expandable nanomaterials with standardized protein building blocks. *bioRxiv*, 2023.2006.2009.544258 (2023).

157. Lee S*, et al.* Design of four component T=4 tetrahedral, octahedral, and icosahedral protein nanocages through programmed symmetry breaking. *bioRxiv*, 2023.2006.2016.545341 (2023).

158. Wicky BIM*, et al.* Hallucinating symmetric protein assemblies. *Science* **378**, 56-61 (2022).

159. Lutz ID*, et al.* Top-down design of protein architectures with reinforcement learning. *Science* **380**, 266-273 (2023).

160. Ingraham J*, et al.* Illuminating protein space with a programmable generative model. *bioRxiv*, 2022.2012.2001.518682 (2022).

161. Levy ED, Erba EB, Robinson CV, Teichmann SA. Assembly reflects evolution of protein complexes. *Nature* **453**, 1262-U1266 (2008).

162. Zlotnick A. Are weak protein-protein interactions the general rule in capsid assembly? *Virology* **315**, 269-274 (2003).

163. Bennett NR*, et al.* Improving de novo protein binder design with deep learning. *Nat Commun* **14**, (2023).

164. Kufareva I, Abagyan R. Methods of protein structure comparison. *Methods Mol Biol* **857**, 231-257 (2012).

165. Zhang Y, Skolnick J. Scoring function for automated assessment of protein structure template quality. *Proteins-Structure Function and Bioinformatics* **57**, 702-710 (2004).

166. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res* **33**, 2302-2309 (2005).

167. Basu S, Wallner B. DockQ: A Quality Measure for Protein-Protein Docking Models. *Plos One* **11**, (2016).

168. Canterakis N. 3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition. In: *In 11th Scandinavian Conf. on Image Analysis*) (1999).

169. Novotni M, Klein R. 3D Zernike descriptors for content based shape retrieval. In: *Proceedings of the eighth ACM symposium on Solid modeling and applications*) (2003).

170. Cheng J*, et al.* Accurate proteome-wide missense variant effect prediction with AlphaMissense. *Science* **381**, eadg7492 (2023).

171. Dill KA, Ozkan SB, Shell MS, Weikl TR. The protein folding problem. *Annu Rev Biophys* **37**, 289-316 (2008).