RWTH AACHEN UNIVERSITY

Computational Design and Cell-Free Expression of Coat Protein Subunits for the Functional Assembly into Potato Virus X-like Particles

Thesis
in Molecular and Applied Biotechnology (B.Sc.)
at RWTH Aachen University

Submitted on: June 30, 2025
by: Kilian Mandon
1. Appraiser: Prof. Dr. Stefan Schillberg
2. Appraiser: Jun. Prof. Dr. Anna Matuszyńska

RWTH Aachen University

Contents

1 Introduction **3**

I Computational Modeling **3**

2 Symmetry Analysis and Model Building **3**

3 Sequence Design with ProteinMPNN **6**

4 Backbone Design with RFdiffusion **9**

5 Evaluation with AlphaFold **12**

6 Evaluation with GROMACS **14**

II Experimental Evaluation **14**

7 Materials **14**
7.1 Laboratory Equipment . . . 14
7.2 Chemicals . . . 14
7.3 Media, Buffers, and Solutions . . . 14
7.4 Reaction Kits . . . 16
7.5 Enzymes . . . 16
7.6 Plasmids . . . 16
7.6.1 pLenEx-Strep-eYFP . . . 16
7.6.2 pLenEx-d29-A, pLenEx-S-Tag-A, pLenEx-S-Tag-B . . . 17
7.6.3 pLenEx-PVX . . . 17
7.7 Antibodies . . . 17
7.8 Synthetic Oligonucleotides . . . 17
7.9 Synthetic Genes . . . 18
7.10 Organisms . . . 18

8 Methods **18**
8.1 DNA Cloning . . . 18
8.1.1 PCR Amplification of Insert DNA . . . 18
8.1.2 Plasmid Restriction Digest . . . 18
8.1.3 Agarose Gel Electrophoresis and DNA Recovery . . . 19
8.1.4 Gibson Assembly . . . 19
8.1.5 Transformation into Competent Cells . . . 20
8.1.6 Colony PCR . . . 20
8.1.7 Plasmid Mini-Preparation and Sequencing . . . 20
8.1.8 Plasmid Midi-Preparation . . . 21
8.2 Protein Expression and Purification . . . 21
8.2.1 Cell-Free Protein Expression . . . 21
8.2.2 Protein Purification Using Capto Core 700 . . . 22
8.3 Protein Analysis . . . 22
8.3.1 SDS-PAGE . . . 22

8.3.2 Coomassie Staining 23
8.3.3 Western Blot 23
8.3.4 ELISA 24
8.3.5 Electron Microscopy 24

9 Results **24**
9.1 DNA Cloning 24
9.2 Protein Expression 24
9.2.1 Coomassie Staining 24
9.2.2 Western Blot 24
9.2.3 ELISA 24
9.2.4 Electron Microscopy 24

Appendix A: Supplementary Figures **25**

Appendix B: Algorithms **25**

Appendix C: Synthetic Sequences **27**

1 Introduction

Part I
Computational Modeling

2 Symmetry Analysis and Model Building

The structure of PVX was determined by [4], up to a resolution of 2.2 Å. The structural data was made available through the PDB, as a file containing 13 consecutive protein subunits, forming one-and-a-half cycles of the helix.

The following chapters require a flexible way to use this symmetry, such as the ability to generate different configurations of monomers (e.g. a 3×3 neighborhood of monomers), or the ability to dynamically enforce this symmetry during symmetry-guided prediction with AlphaFold (Section ...) or symmetry-guided design with RFdiffusion (Section ...). Therefore, this section discusses the computation of the symmetry relationship between consecutive monomers, and how it can be applied to generate new configurations of monomers.

Let $\{\vec{\mathbf{r}}_{j,i}^{\text{original}}\}$ denote the backbone atom positions of chain j in the original PDB file, and let $\{\vec{\mathbf{r}}_{j}^{\text{original}}\}$ be their arithmetic mean.

We choose $T_0 = (I_3, \vec{\mathbf{r}}_A^{\text{original}})$ as our new origin, centered on chain A. The backbone atom coordinates in this frame are denoted by $\vec{\mathbf{r}}_{j,i}$, and we have

$$\vec{\mathbf{r}}_{j,i} = T_0^{-1} \circ \vec{\mathbf{r}}_{j,i}^{\text{original}} = \vec{\mathbf{r}}_{j,i}^{\text{original}} - \vec{\mathbf{r}}_A^{\text{original}} \tag{1}$$

The frames of all other chains in these coordinates are computed as the optimal rigid body transform to align the chain with A. That is,

$$T_j = \underset{T\in\mathrm{SE}(3)}{\arg\min} \sum_i \left\| T \circ \vec{\mathbf{r}}_{A,i} - \vec{\mathbf{r}}_{j,i} \right\|^2 \tag{2}$$

Using the Kabsch algorithm [5], T_j can be computed as $T_j = (R_j, \vec{\mathbf{t}}_j)$, where

$$\vec{\mathbf{t}}_j = \vec{\mathbf{r}}_j - R_j\vec{\mathbf{r}}_A = \vec{\mathbf{r}}_j \tag{3}$$

since $\vec{\mathbf{r}}_A = \vec{\mathbf{0}}$, and $R_j \in \mathrm{SO}(3)$ minimizes

$$\sum_i \left\| R_j(\vec{\mathbf{r}}_{A,i} - \vec{\mathbf{r}}_A) - (\vec{\mathbf{r}}_{j,i} - \vec{\mathbf{r}}_j) \right\| \tag{4}$$

Following the Kabsch Algorithm, R_j can be computed via the singular value decomposition

$$(\vec{\mathbf{r}}_{A,i} - \vec{\mathbf{r}}_A)^T \cdot (\vec{\mathbf{r}}_{j,i} - \vec{\mathbf{r}}_j) = U\Sigma V^T \tag{5}$$

as

$$R_j = V \cdot \mathrm{diag}(1, 1, d) \cdot U^T \tag{6}$$

where $d = \det(U)\det(V)$ corrects for a potential reflection in the orthogonal matrices U and V.

With all frames T_j expressed in the same coordinate system, we can compute the relative transform

$$T_{j\to j+1} = (R_{j\to j+1}, \vec{\mathbf{t}}_{j\to j+1}) = T_j^{-1} \circ T_{j+1} \tag{7}$$

Given the symmetry of the viral coat structure, these transforms are expected to be equal. The average relative transform $T_R = (R_R, \vec{\mathbf{t}}_R)$ is computed by choosing $\vec{\mathbf{t}}_R$ as the mean over $\{\vec{\mathbf{t}}_{j\to j+1}\}$ and choosing $R_R \in \mathrm{SO}(3)$ as the rotation matrix closest to the average over all $R_{j\to j+1}$, that is $R_R = UV^T$ where $U\Sigma V^T = \frac{1}{n}\sum_j R_{j\to j+1}$ [7] (given the similarity of the $\{R_{j\to j+1}\}$, no reflection can arise by continuity).

The individual rotations $R_{j\to j+1}$ had standard deviation $\Delta R_R = 0.004\,\mathrm{rad}$ in geodesic distance, and the individual translations had standard deviation $\Delta \mathbf{t}_R = 0.04\,\text{Å}$. R_R closely resembles a pure rotation around the z-axis $R_Z(\theta)$, with an angle of $\theta = -0.707\,\mathrm{rad}$. The deviation is $d(R_R, R_Z(\theta)) = 0.005\,\mathrm{rad}$. This value of θ corresponds to a left-handed helix with 8.89 subunits per turn. The computed rise is $\mathbf{t}_z = 3.87\,\text{Å}$ per subunit, resulting in a helical pitch (rise per turn) of 34.4 Å. These values are mostly consistent with the ones stated in [4] (rise 3.96 Å, rotation of 0.707 rad, 8.9 copies per turn, helical pitch 35.2 Å). However, the authors emphasize the slight difference in the helical pitch of 35.2 Å compared to that of similar flexible filamentous plant viruses (PepMV, BaMV, and PapMV), for which the helical pitch ranges from 34.3 Å to 34.6 Å. According to the calculations above, the helical pitch in the PDB entry (which the authors produced through multiple cycles of real space refinement) differs from the original helical parameters fitted to the cryo-EM data and falls into the range of the other plant viruses, thereby potentially diminishing the significance of the reported pitch deviation.

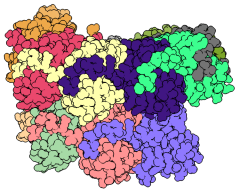
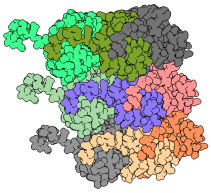
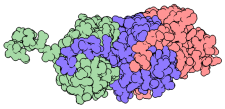
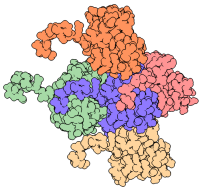
Given the relative transform T_R, model coordinates can be reconstructed based on the coordinates of the monomer A according to

$$\vec{\mathbf{r}}_{j,i}^{\,\text{original}} = T_0 \circ T_R^j \circ \vec{\mathbf{r}}_{A,i} \tag{8}$$

Using equation 8, four different configurations of monomers are generated and used throughout the following sections (Table 1). A helical configuration consisting of thirteen consecutive monomers, a three-by-three neighborhood of nine monomers, a trimer consisting of three consecutive monomers, and a pentamer consisting of five monomers aranged in a cross-shape.

Despite the small standard deviation of T_R, the deviation of individual atom positions in the helical thirteen-monomer reconstruction compared to the data from the pdb entry reaches up to 0.8 Å. This is due to lever effects caused by small deviations in the rotation. The difference in structure introduces no new clashes, but slightly reduces the contacts by 2 %, as computed with ChimeraX [6].

Table 1: Visualization and chain indices of different monomer configurations, generated based on the average relative transform T_R. The blue chain has index 0, the coordinates for the other chains are computed as $T_R^j \circ \vec{\mathbf{r}}_{A,i}, j \in I$. The generated monomer configurations will be used to create inputs for the algorithms in the following sections.

Type	**Indices**	**Visualization**
Helical	$I = \{0, ..., 12\}$	
3x3	$I = \{0, \pm 1, \pm 8, \pm 9, \pm 10\}$	
Trimer	$I = \{0, \pm 1\}$	
Pentamer	$I = \{0, \pm 1, \pm 9\}$	

3 Sequence Design with ProteinMPNN

ProteinMPNN [3] is a deep learning model for protein sequence design, capable of creating de-novo designs of proteins that fold into a desired shape or bind to specific targets. The algorithm can create sequences for monomers, heterooligomers, and homooligomers.

The sequence is designed based on a protein backbone as input, that is the position of all backbone atoms of one or multiple chains. The underlying algorithm uses a Message Passing Neural Network (MPNN), a graph-based machine learning model. Each residue in the protein is encoded as a vertex in the graph, and edges are drawn up from each residue to its 48 closest neighbors. Vertex embeddings are initialized as 0 vectors, while the initial edge embeddings are computed based on the distances between the backbone atoms of the residue pair and the difference of their residue indices. After the computation of the initial feature embeddings, ProteinMPNN follows an encoder-decoder architecture, in which the encoder updates the edge and vertex embeddings based on their neighborhood, wheras the decoder uses the embeddings computed by the encoder to predict the amino acid type for each residue. The decoder works in an autoregressive fashion by choosing a random order for decoding the individual residues, then predicting their residue type one-by-one with knowledge of all already predicted residues. Concretely, the algorithm predicts logits $\{\ell_i\}$ for each amino acid and chooses it from a softmax distribution according to

$$P(a_i) = \frac{\exp\left(\frac{\ell_i}{\tau}\right)}{\sum_{j=1}^{20} \exp\left(\frac{\ell_j}{\tau}\right)}$$

Here, $\tau > 0$ denotes a chosen temperature constant in the softmax distribution. For $\tau \to \infty$, the distribution is almost uniform, while for $\tau \to 0$ the amino acid with the highest predicted logit is chosen. The distribution can be biased by adding to the logits before sampling. For homooligomers, the logits of identical residues in different monomers are averaged and only one amino acid is sampled from the distribution for all of them.

In this work, all sequences used in computational and experimental evaluation are generated using ProteinMPNN. The input structure is either chosen as the backbone structure of the wildtype, thereby generating alternative sequences for the structure, or a generated artificial backbone as described in Section 4. Of particular note is the choice of the input structure: The helical virus particle consists of approximately 1300 monomers [4], and truncation to a smaller number will lead to an incorrect neighborhood during featurization for newly exposed residues.

However, a modification to the original ProteinMPNN algorithm can circumvent this by allowing sequence prediction for a theoretical infinite extension of a symmetric homooligomer. In ProteinMPNN, feature initialization is solely dependent on the relative neighborhood of each residue, meaning that initialization is identical for all corresponding residues in a symmetric homooligomer. Further, the message-passing algorithm in the network conserves this equivariance. Therefore, a theoretical infinite extension of the homooligomer can be simulated by remapping of interchain edges to the corresponding residue in the same chain (Figure 1), thereby reducing the input to a single monomer.

When testing this new algorithm for different helical viruses, the Graph Reduction procedure showed no significant improvement compared to prediction based on

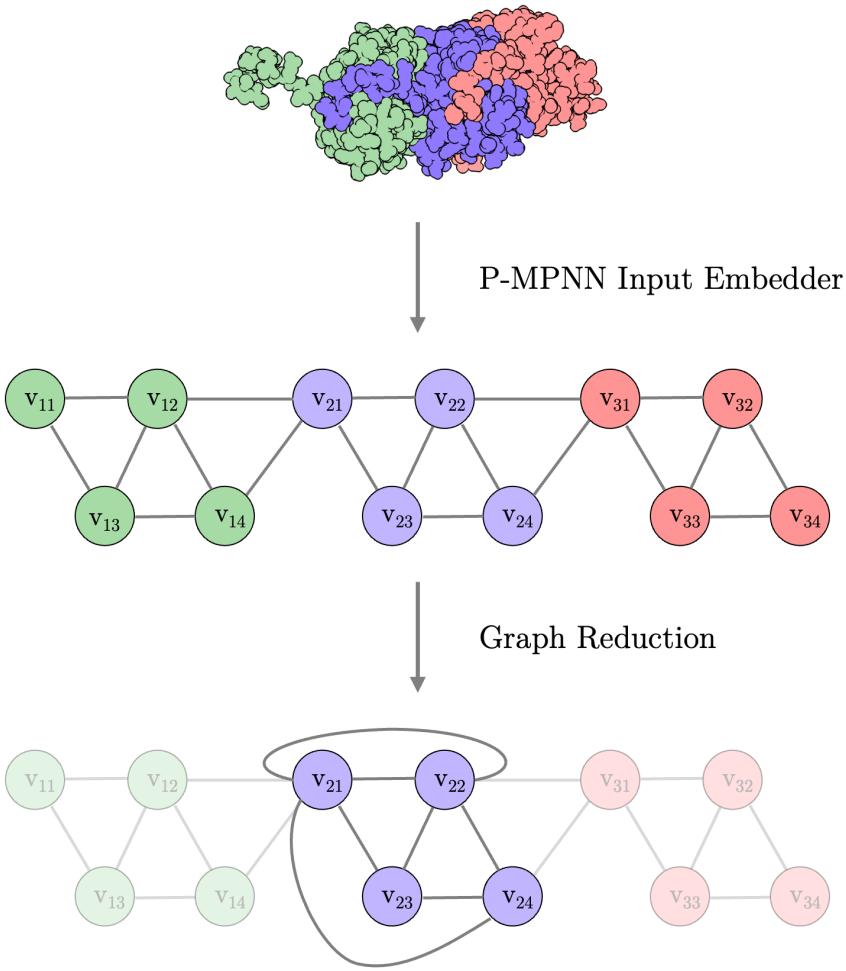


Figure 1: Graph Reduction procedure for symmetric homooligomers. After the default graph initialization from ProteinMPNN, one of the monomers is chosen as the reference monomer. Edges going out from it to other monomers are remapped to the corresponding residue in itself. Afterward, vertices and edges of the non-reference monomers are discarded.

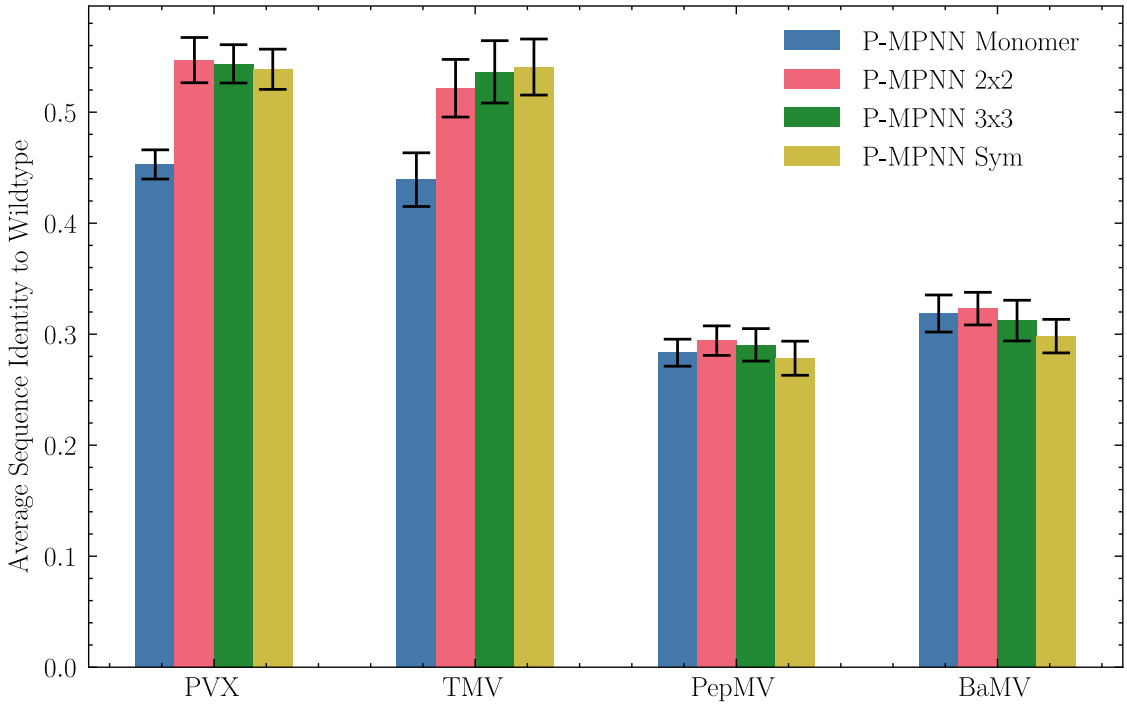


Figure 2: Sequence recovery by ProteinMPNN for different input configurations. The input was chosen as either a single monomer, a 2x2 neighborhood, a 3x3 neighborhood, or a symmetry-preserving graph reduction, modeling a theoretical infinite neighborhood. For each of the four targets Potato Virus X (PVX), Tobacco Mosaic Virus (TMV), Pepino Mosaic Virus (PepMV) and Bamboo Mosaic Virus (BaMV), each model was evaluated 50 times using random decoding orders and a sampling temperature $\tau \to 0$. The errorbars indicate the standard deviation over the repeated evaluation.

a 2x2 neighborhood or a 3x3 neighborhood of monomers (Figure 2). For PVX and Tobacco Mosaic Virus (TMV), the three multimeric inputs (2x2 / 3x3 / infinite neighborhood) performed better than prediction based on a sole monomer, while no such improvement was observed for Pepino Mosaic Virus (PepMV) and Bamboo Mosaic Virus (BaMV) where all methods had similar sequency recovery rates. These results suggest that for the tested proteins, the incorrect neighborhood for small crops doesn't lead to an increased call of wrong amino acids in the aggregated logits. The newly developed infinite symmetry approach performs en par with 2x2 or 3x3 neighborhood prediction, but lowers the amount of required compute to that of a single monomer. However, it is to note that compute cost is generally not a concern when running ProteinMPNN due to its low complexity.

ProteinMPNN was used to generate sequences based on the wildtype backbone structure of PVX using the introduced Graph Reduction technique to model an infinite symmetry and a sampling temperature of $\tau \to 0$, e.g. argmax sampling. Sequences where generated with varying bias b towards the wildtype sequence, that is by increasing the logit of the residue that's present in the wildtype structure by b before sampling the amino acid. For each of the bias values $b \in \{0, 1, 2, 2.5\}$, five sequences. The sequence identity of the generated sequences to the wildtype was about 0.54 (bias 0), 0.73 (bias 1), 0.88 (bias 2) and 0.94 (bias 2.5). The generated sequences where further analyzed as described in the sections 5 and 6 before selecting

some for experimental evaluation.

4 Backbone Design with RFdiffusion

RFdiffusion is a generative machine learning model for protein backbone design. It can be run in different modes to accomplish several tasks such as unconditional monomer generation, protein binder design, scaffolding around a fixed motif, or design of symmetric oligomers (Figure 3), the latter being the most relevant for this work. In practice, RFdiffusion is commonly used together with ProteinMPNN, where RFdiffusion generates synthetic backbone structure and ProteinMPNN tries to realize this backbone with a synthetic amino acid sequence.

Generation by RFdiffusion is performed through a reverse Riemannian diffusion process on the manifold SE(3). Compared to other diffusion-based algorithms like AlphaFold3 (Section 5), RFdiffusion doesn't operate on the atom coordinates using standard euclidean diffusion, but diffuses the backbone transforms instead. However, it converts the transforms from and to atom coordinates in each iteration. For unconditional generation, the model starts with randomly initialized backbone coordinates and creates a based solely on a specified number of residues. For the creation of symmetric oligomers, the user specifies a set of transforms $\mathfrak{R} = \{R_k\}_{k=1}^{K} \in \mathrm{SO}(3)$ that define the symmetry. The final protein will satisfy $x^{(k)} = R_k x^{(1)}$, where $x^{(k)}$ denotes the coordinates of the k-th monomer. This is achieved by explicitly setting the coordinates as such after initialization and in each further iteration (algorithm 1).

Algorithm 1 Generation of symmetric oligomers

def SampleSymmetric($M, \mathfrak{R} = \{R_k\}_{k=1}^{K}$):

\# RFdiffusion generation of oligomer with symmetry $\mathfrak{R}$

$x^{(T,1)} = \mathrm{SampleReference}(M)$

for all $t = T, \ldots, 1$ **do**

\# Symmetrize chains

$X^{(t)} = [R_1 x^{(t,1)}, \ldots, R_K x^{(t,1)}]$

$\hat{X}^{(0)} = \mathrm{RFdiffusion}(X^{(t)})$

$[x^{(t-1,1)}, \ldots, x^{(t-1,K)}] = \mathrm{ReverseStep}(X^{(t)}, \hat{X}^{(0)})$

end for

return $\hat{X}^{(0)}$

In the original RFdiffusion code, only point group symmetries are supported, that is symmetries that satisfy $\{x^{(1)}, ..., x^{(K)}\} = \{R_j x^{(1)}, ..., R_j x^{(K)}\}$ for each R_j, up to reordering of the monomers. Technically, the code could work on general euclidean transforms $T_j \in \mathrm{SE}(3)$ (such as the transforms from Section 2 specifying the symmetry of PVX) as well. However, there are certain drawbacks in doing so. The positions in early steps in the diffusion process follow a Gaussian distribution. While the rotations by point group symmetries conserve that distribution, general

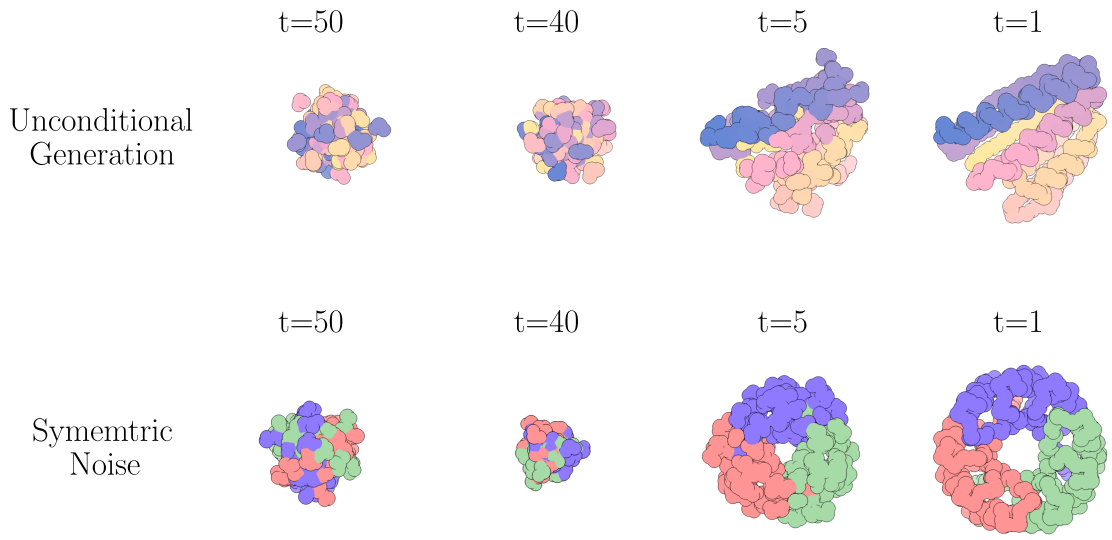


Figure 3: RFdiffusion trajectories for unconditional generation and symmetric noise. For unconditional generation, RFdiffusion samples the initial positions independently from a Gaussian distribution and generates the protein without any constraints. For the generation of oligomers with a specific symmetry, RFdiffusion only samples coordinates for one monomer and initializes the other coordinates by applying the respective symmetry transform to the coordinates of that reference monomer. In each diffusion step, this is repeated to enforce the symmetry.

euclidean transforms don't. This means that the positions that are fed into the noise prediction network in the diffusion process don't follow the distribution the model is trained on. Further, the authors of RFdiffusion observed that for point group symmetries, the noise prediction model conserves the symmetry almost perfectly. Due to this, the explicit symmetrization in each iteration is generally not necessary and barely affects the trajectory, if the initial noise is symmetrized. This arises from the equivariance of the SE(3)-transformer architecture used in RFdiffusion. Using the symmetry transforms for PVX as evaluated in Section 2, the atom positions in early stages of the diffusion process don't follow a Gaussian distribution, and same-seeded trajectories using either full symmetry enforcement or only initially symmetrized noise differ strongly (Figure 4). Rather, for initial symmetrization, the atoms quickly collapse to a Gaussian-like distribution, before spreading out again to form the final multimer.

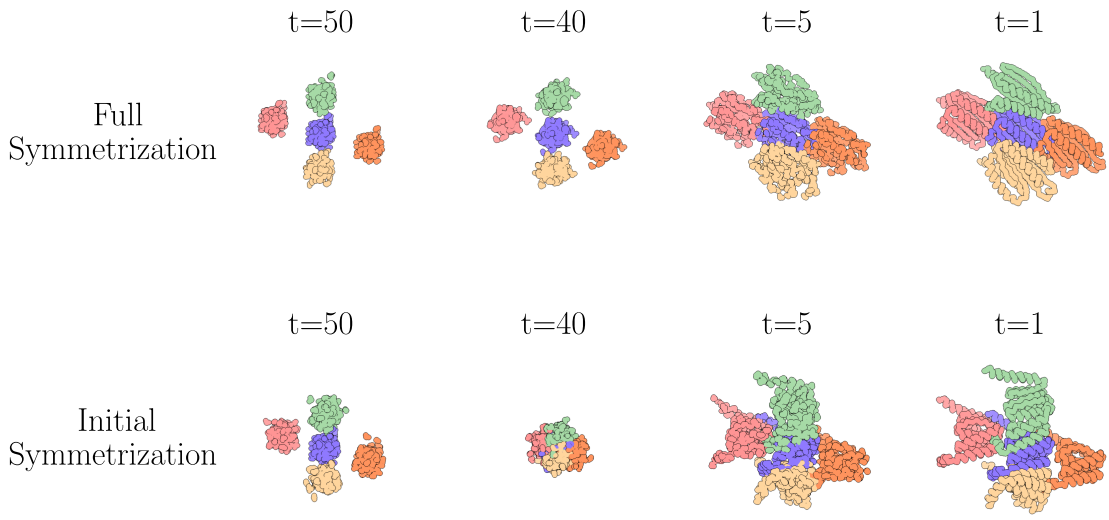


Figure 4: RFdiffusion trajectories for full and initial symmetry enforcement.

RFdiffusion with full symmetry enforcement of the PVX symmetry was used to generate backbone structures for further testing. In total, five different backbone structures where designed, and ProteinMPNN was used as described in Section 3 to generate five sequences for each of them. Additionally, backbone structures were generated using partial denoising, where only a limited amount of noise was added to the wild type structure before denoising again. This was done using 5, 10, 15, and 20 noise steps in RFdiffusion. For each of these noise levels, three denoised backbones were generated using RFdiffusion, each realized by three sequences through ProteinMPNN. The de novo generated backbones typically consist of a simple structure of alpha helices and beta sheets (Figure 5). The sequences were further evaluated using the methods described in sections 5 and 6.

Possibly due to the aforementioned incongruities in running the algorithm with euclidean transforms, backbone structures generated for the PVX symmetry tend to have structural violations, in particular interchain clashes.

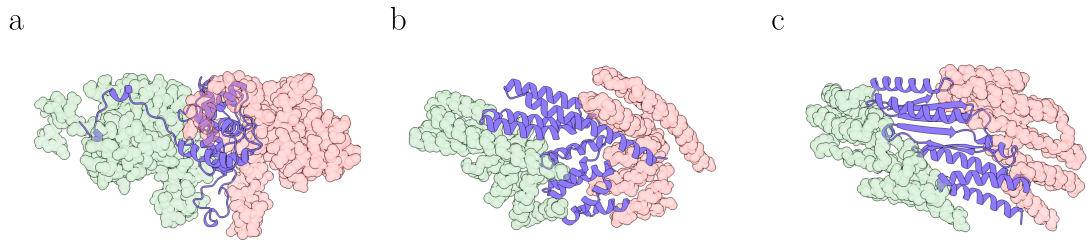


Figure 5: RFdiffusion examples.

5 Evaluation with AlphaFold

Despite outstanding performance of RFdiffusion and ProteinMPNN for de novo protein design, the success rate is often too low for a small-scale experimental evaluation. In experiments by Watson et al. [8], for the task of symmetric oligomer designs, 87 out of 608 designs showed an oligomerization state consistent with the design models. In light of this, methods for in silico assessment of protein designs were established to improve the chances for successful designs.

For the task of binder design, Bennett et al. managed to increase the success rate of binder design nearly 10-fold using metrics based on AlphaFold 2 [2]. In their design assays, a low Cα RMSD and a low predicted aligned error (pAE) between inter-chain residue pairs was predictive of binder success. Unfortunately, AlphaFold 2 fails to predict the multimeric structure of the wild type. Even using the "AF2 initial guess" method [2] of providing the expected backbone structure to the model through the recycling embedder was unsuccessful in recovering the prediction.

The recently developed model AlphaFold 3 [1] performs better on the prediction of the wild type, but still only makes a prediction with Cα RMSD of 5 Å or less in 8 % of the evaluations (Figure 7). Since AlphaFold 3 is indeterministic, repeated runs result in different outcomes. However, the architecture for structure prediction in AlphaFold 3 is largely different from AlphaFold 2, replacing the structure module with a diffusion algorithm. As seen in Section 4, diffusion algorithms allow for changes to the denoising process to guide the prediction, such as a symmetry constraint.

While both RFdiffusion and AlphaFold 3 use diffusion, their exact implementations vary, requiring additional considerations when transferring the symmetrization process used in RFdiffusion to AlphaFold 3. In particular, the diffusion trajectories in AlphaFold 3 are not scaled to unit variance and the model changes its position and orientation throughout the process (Figure 6). This motion of the model can be tracked by using a reference frame $T_{\text{ref}} = (R_{\text{ref}}, \vec{\mathbf{t}}_{\text{ref}})$ and enforcing the symmetry in that frame as

$$\vec{\mathbf{x}}^{(j)} = T_{\text{ref}} \circ T_j \circ T_{\text{ref}}^{-1} \circ \vec{\mathbf{x}}^{(1)} \tag{9}$$

Motion of the model happens in three stages of the diffusion sampler in AlphaFold 3: First, the function CentreRandomAugmentation recenters the prediction before applying a random rotation and translation to the model. Second, Gaussian noise is added to the model in each iteration, potentially shifting it. Third, the prediction by the denoiser can be shifted, resulting in a translation of the model when applying the denoising step. These motions do not occur in RFdiffusion. Since the algorithm is SE(3) invariant, it does not require augmentation. No noise is added, and the prediction is aligned to the current model before updating it.

To account for this, the motion by the function CentreRandomAugmentation can be applied to the reference frame T_{ref} as well, and shifts to the model can be considered by setting the translation $\vec{\mathbf{t}}_{\text{ref}}$ to the center of the reference monomer in each iteration. Further, the prediction by the denoiser can be shifted to match the center of the current model before applying the symmetry. The details of the implementation are outlined in Algorithm 3. In this work, the symmetry constraint was applied to the initial noise and the denoised prediction. Symmetrization of the current model at the start of each iteration, as is done in RFdiffusion, is likely to be similarly effective. The denoised prediction could also lead to a slight rotation of the

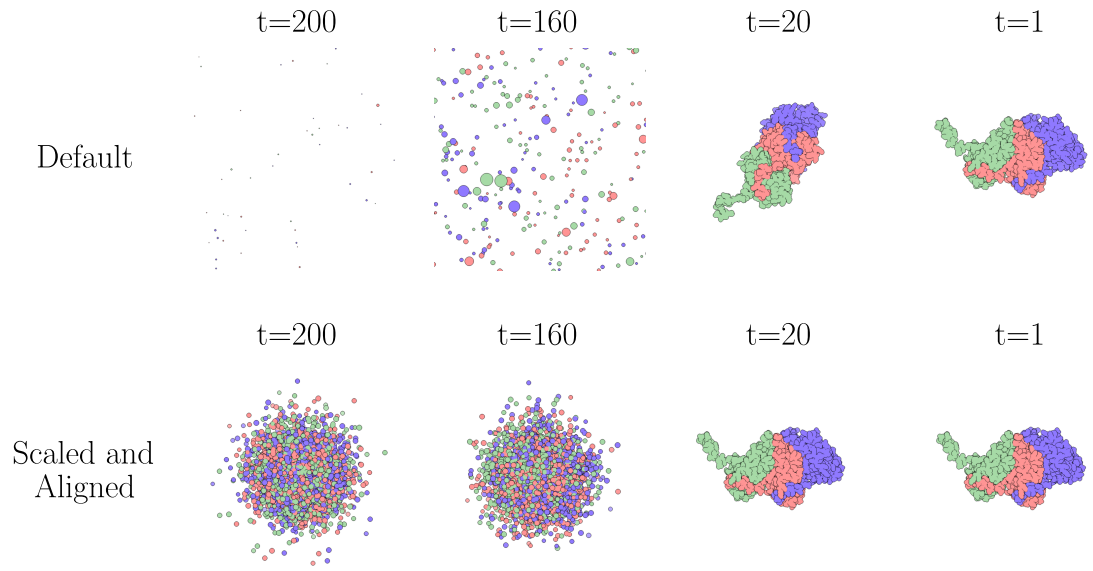


Figure 6: Diffusion Trajectories of AlphaFold3 with symmetry enforcement.

model over time. This can be considered by rotating the reference frame towards the best alignment of the current model with the expected backbone coordinates. Notably, the atom coordinates in early steps of the diffusion process have a standard deviation that is significantly larger than the translation in the symmetry transforms of PVX. Due to this, symmetrization in AlphaFold 3 does not strongly affect the distribution, as it did for RFdiffusion (Section 4).

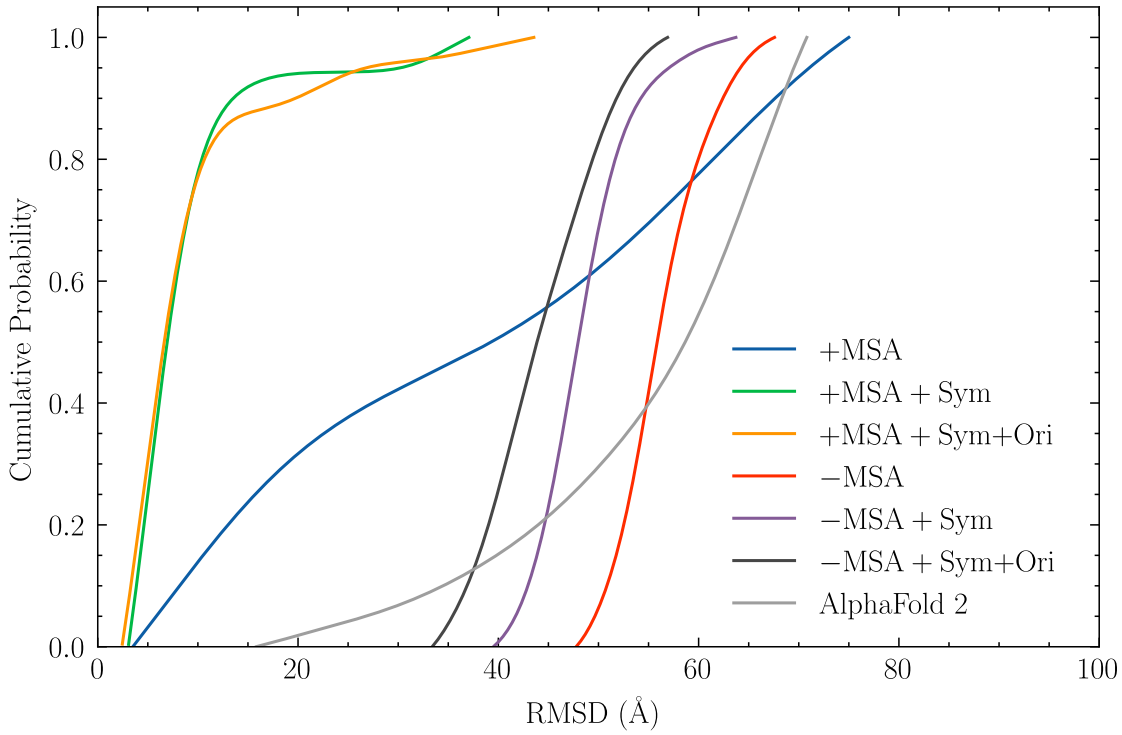


Figure 7: Comparison of Cα RMSD of AlphaFold 3 on PVX using different variants of the algorithm.

6 Evaluation with GROMACS

Part II
Experimental Evaluation

7 Materials

7.1 Laboratory Equipment

7.2 Chemicals

7.3 Media, Buffers, and Solutions

All media, buffers, and solutions that were used in the experiments are shown in table 2. NaOH and HCl were used to establish the pH.

Table 2: Media, buffers, and solutions that were used in this study. Listed are the component types their respective amounts.

Medium/buffer/solution	Component	Amount
agarose gel	Agarose	1.2 % (w/v)
	Ethidium bromide	0.5 µg mL^{-1}

Medium/buffer/solution	**Component**	**Amount**
	in 1x TAE buffer	
AP buffer (pH 9.6)	Tris-HCl	100 mM
	NaCl	100 mM
	$MgCl_2$	5 mM
blocking solution	powedered milk	$40\,g\,L^{-1}$
	in PBS buffer	
Coomassie staining solution	Coomassie Brilliant Blue	$2.5\,g\,L^{-1}$
	Methanol	50 % (v/v)
	Acetic acid	10 % (v/v)
coating buffer (pH 9.6)	—	—
destaining solution	Methanol	5 % (v/v)
	Acetic acid	7.5 % (v/v)
elisa substrate solution	—	—
LB medium	Yeast extract	$5\,g\,L^{-1}$
	Tryptone	$10\,g\,L^{-1}$
	NaCl	171 mM
	(Agar)	$15\,g\,L^{-1}$
LB-Amp medium	Yeast extract	$5\,g\,L^{-1}$
	Tryptone	$10\,g\,L^{-1}$
	NaCl	171 mM
	Ampicillin	$100\,mg\,L^{-1}$
	(Agar)	$15\,g\,L^{-1}$
NBT/BCIP staining solution	NBT	$33.3\,g\,L^{-1}$
	BCIP	$16.5\,g\,L^{-1}$
	in dimethylformamide	
PBS buffer	NaCl	137 mM
KCl	2.7 mM	
Na_2HPO_4	8.1 mM	
KH_2PO_4	1.5 mM	
phosphate buffer (0.01 M, pH 7.2)	Na_2HPO_4	6.67 mM
	NaH_2PO_4	3.33 mM
reducing loading buffer (5x)	Tris-HCl (pH 6.8)	62.5 mM
	Glycerine	$300\,g\,L^{-1}$
	SDS	$40\,g\,L^{-1}$
	β-Mercaptoethanol	10 % (v/v)
	Bromophenol blue	$0.1\,g\,L^{-1}$
Sabu (10x)	Glycine	50 % (v/v)
	Xylene cyanol FF	0.1 % (w/v)
	Bromophenol blue	0.1 % (w/v)
	in 1x TAE buffer (pH 8.0)	
SDS running buffer	Glycine	200 mM
	SDS	$1\,g\,L^{-1}$

Medium/buffer/solution	Component	Amount
	Tris	25 mM
semi-dry transfer buffer	Glycine	391 mM
	Methanol	20 % (v/v)
	Tris	48 mM
TAE buffer	Tris	40 mM
	Acetic acid	0.1 % (v/v)
	EDTA (pH 8.0)	1 mM

7.4 Reaction Kits

The following reaction kits were used in this study:

- Gibson Assembly® Cloning Kit, NEB, Ipswich, MA
- Mix2Seq Kit, Eurofins Genomics, Ebersberg, Germany
- PureYield™ Plasmid Miniprep System, Promega, Madison, WI
- Wizard® SV Gel and PCR Clean-Up System, Promega, Madison, WI
- NucleoBond Xtra Midi kit for transfection-grade plasmid DNA, Macherey-Nagel, Düren, Germany
- ALiCE® for Research kit, LenioBio, Düsseldorf, Germany

7.5 Enzymes

All enzymes used in this study are shown in table 3.

Table 3: Polymerases and restriction enzymes that were used in this study. Listed are the reagent names, the manufacturer, and how the enzymes were used in the experiments.

Reagent	Manufacturer	Use
GoTaq® G2 DNA Polymerase	Promega	Colony PCR
Pfu DNA Polymerase	Promega	Insert Amplification
NcoI-HF®	NEB	Restriction
KpnI-HF®	NEB	Restriction

7.6 Plasmids

7.6.1 pLenEx-Strep-eYFP

The plasmid pLenEx-Strep-eYFP was used for cell-free expression of Strep-eYFP using the ALiCE® expression kit. The plasmid contains an origin of replication for *E. coli*, a gene for ampicillin resistance, and a gene encoding Strep-eYFP. This gene is flanked by 5'- and 3'-UTRs from tobacco mosaic virus (TMV) and under the T7

promoter. The plasmid contains restriction sites for NcoI at the 5' end of Strep-eYFP and for KpnI at the 3' end. In addition to cell-free expression, pLenEx-Strep-eYFP was used for cloning of the genetic elements d29-A, S-Tag-A, and S-Tag-B. The plasmid was provided by the Institute for Molecular Biotechnology.

7.6.2 pLenEx-d29-A, pLenEx-S-Tag-A, pLenEx-S-Tag-B

Like pLenEx-Strep-eYFP, the plasmids pLenEx-d29-A, pLenEx-S-Tag-A, and pLenEx-S-Tag-B contain an origin of replication for *E. coli* and a gene for ampicillin resistance. Instead of Strep-eYFP, the vectors carry the genetic elements d29-A, S-Tag-A, and S-Tag-B, as described in Section 6, respectively. These genes are again flanked by 5'- and 3'-UTRs from TMV and are under control of the T7 promoter and in-between the NcoI and KpnI restriction sites. The plasmids were created based on pLenEx-Strep-eYFP using Gibson Assembly.

7.6.3 pLenEx-PVX

The plasmid pLenEx-PVX carries the same genetic elements as pLenEx-Strep-eYFP, except for the Strep-eYFP gene, which is instead replaced by a gene encoding for the PVX coat protein. The plasmid was provided by the Institute for Molecular Biotechnology.

7.7 Antibodies

All antibodies that were used in this study are listed in table 4. The antibodies were used both for Western Blotting and for ELISA assays.

Table 4: Antibodies that were used in this study. Listed are the characteristics of the antibody and its manufacturer.

Antibody	Type
Rabbit-Anti-PVX	Polyclonal rabbit IgG
S-peptide Epitope Tag Monoclonal Antibody	Monoclonal mouse IgG
Goat-Anti-Rabbit FC AP	Polyclonal goat IgG, alkaline phosphatase cor
Goat-Anti-Mouse FC AP	Polyclonal goat IgG, alkaline phosphatase cor

7.8 Synthetic Oligonucleotides

All syntehtic oligonucleotides that were used in this study are shown in table 5.

Table 5: Synthetic oligonucleotides that were used in this study. Regions overlapping with the amplified genes are displayed in uppercase.

Name	Sequence (5' → 3')	Use
d29-A fwd	acattttacattctacaactaccATGGCTT CTGGCTTATTCACCATACCTG	Insert amplification
d29-A rev	ccaaaccagaagagcttggtaccTTAAGGG GGGGGAATGGTCAC	Insert amplification
S-Tag-A fwd	acattttacattctacaactaccatggcTA AAGAAACAGCCGCCGCTAAATTC	Insert amplification
S-Tag-B fwd	acattttacattctacaactaccatggctA AGGAGACTGCTGCAGCCAAG	Insert amplification
S-Tag-B rev	ccaaaccagaagagcttggtaccTTAAGCT GCGGGTATGTGTATGATTC	Insert amplification
pLenSeq fwd	% TODO: sequence missing	Sequencing
pLenSeq rev	% TODO: sequence missing	Sequencing

7.9 Synthetic Genes

Genes for the constructs S-Tag-A and S-Tag-B were ordered from The exact sequences can be found in 9.2.4.

7.10 Organisms

NEB® Turbo Competent *E. coli* (High Efficiency) was used for cloning of the plasmids pLenEx-d29-A, pLenEx-S-Tag-A, and pLenEx-S-Tag-B.

8 Methods

8.1 DNA Cloning

8.1.1 PCR Amplification of Insert DNA

Using the synthetic genes S-Tag-A and S-Tag-B (Subsection 7.9) and the primer pairs d29-A fwd / d29-A rev (with template S-Tag-A), S-Tag-A fwd / d29-A rev (with template S-Tag-A), and S-Tag-B fwd / S-Tag-B rev (with template S-Tag-B), polymerase chain reactions (PCRs) were conducted for amplification of the inserts. The primers also introduced overlapping regions with the plasmid pLenEx-Strep-eYFP for the following Gibson Assembly, and in the case of d29-A fwd, a start codon. The PCR was conducted with a Pfu polymerase. The composition of the PCR mix is listed in Table 6. The PCR reaction was conducted in a thermocycler. The exact program is described in Table 7. After running the PCR, the products were frozen at −20 °C until further processing.

8.1.2 Plasmid Restriction Digest

The plasmid pLenEx-Strep-eYFP was digested using the restriction enzymes NcoI-HF and KpnI-HF. Therefore, 10 µg of the plasmid, 5 µL CutSmart® Buffer, 1 µL of

Table 6: Composition of the PCR Mix for Insert Amplification.

Component	Amount
MQ-H_2O	38 µL
10x Pfu Buffer	5 µL
dNTPs (10 mM)	2 µL
Forward Primer (10 µM)	2 µL
Reverse Primer (10 µM)	2 µL
Pfu DNA Polymerase (TODO: Lookup activity)	0.5 µL
DNA template (TODO: Lookup concentration)	5 µL

Table 7: PCR Program used for Insert Amplification.

Step	Temperature (°C)	Time
Initial Denaturation	94	4 min
Repeat for 30 cycles:		
Denaturation	94	30 s
Annealing	57	30 s
Elongation	72	100 s
Final Elongation	72	5 min
Storage	8	∞ (hold)

NcoI-HF (TODO: Lookup activity), and 1 µL of KpnI-HF (TODO: Lookup activity) were added to MQ-H_2O up to a total volume of 50 µL. The mixture was incubated at 37 °C for three hours and thereafter frozen at −20 °C until further processing.

8.1.3 Agarose Gel Electrophoresis and DNA Recovery

Agarose gel electrophoresis was used to validate and purify the restricted plasmid and the PCR products. 50 µL of each sample were mixed with 5 µL 10x Sabu and applied to the gel, splitting the sample into 25 µL per track. The gels were run at either 100 V for about 45 minutes or at 120 V for about 60 minutes, depending on the size of the gel. After the gel ran through, the samples were cut out under illumination from a UV light source. The DNA was recovered from the gel samples using the Wizard® SV Gel and PCR Clean-Up System. The steps were carried out following the manufacturer's protocol. DNA concentrations of each sample were determined using the NanoDrop™ One.

8.1.4 Gibson Assembly

A Gibson Assembly using 50 ng of purified linear vector DNA was conducted to create the plasmids pLenEx-d29-A, pLenEx-S-Tag-A, and pLenEx-S-Tag-B. The required mass of insert DNA was calculated using Equation 10.

$$\text{Mass}_{\text{insert}} = \left(\frac{\text{desired molar ratio}}{1}\right) \times \text{Mass}_{\text{vector}} \times \left(\frac{\text{Length}_{\text{insert}}}{\text{Length}_{\text{vector}}}\right) \tag{10}$$

The molar ratio was chosen as 2. Given the plasmid length of 2173 bp, and the insert lengths of 679 bp, 802 bp, and 802 bp for the three inserts d29-A, S-Tag-A, and S-Tag-B respectively, the insert DNA masses were calculated as 31.2 ng for d29-A and 37 ng for S-Tag-A and S-Tag-B. The plasmid and inserts and 10 µL Gibson Assembly® Master Mix were added to MQ-H_2O to a total volume of 20 µL and incubated at 50 °C for one hour.

8.1.5 Transformation into Competent Cells

Transformation of the assembled plasmids into NEB® Turbo Competent *E. coli* cells was carried out using the manufacturer's High Efficiency Transformation Protocol, using a reduced volume of cells compared to the original protocol.

After thawing the cells on ice for 10 minutes, the cells were gently mixed and 20 µL of the cells were transferred to a reaction tube on ice. TODO 0 ng of plasmid DNA were added to the cell mixture and carefully flicked to mix cells and DNA. The mixture was placed on ice for 30 minutes. Afterward, a heat shock at exactly 42 °C was conducted for 30 seconds. The cells were thereafter placed on ice for 5 minutes. Afterward, 950 µL of room temperature salt medium was added to the mixture. The cells were rotated at 37 °C for 60 minutes, during which LB-Amp selection plates were warmed to 37 °C. The cells were mixed thoroughly by flicking the tube and inverting it. Afterward, 100 µL were applied to a selection plate and incubated overnight at 37 °C.

8.1.6 Colony PCR

After incubation overnight, 9 colonies of each construct were picked for Colony PCR. The master mix for the PCR was created following the composition in Table 8.

Table 8: PCR Master Mix Composition

Component	Amount
MQ-H_2O	15 µL
5x Green GoTaq® Reaction Buffer	4 µL
dNTPs (10 mM)	0.4 µL
Forward Primer	0.2 µL
Reverse Primer	0.2 µL
Taq DNA Polymerase	0.2 µL

For bidirectional sequencing sample, 20 µL of the master mix were transferred into PCR tubes. The colonies were picked using sterile toothpicks and applied to an LB-Amp reference plate. Afterward, they were placed into the PCR tubes for about 30 seconds. The reference plate was incubated at 37 °C overnight. The PCR was run using the program from Table 7 and was followed by an agarose gel electrophoresis as previously described for analysis of the PCR products.

8.1.7 Plasmid Mini-Preparation and Sequencing

For ebidirectional sequencing construct, two clones showing successful amplification during the colony PCR were selected for a plasmid mini preparation and sequencing.

6 mL of LB-AMP medium were inoculated and incubated overnight at 37 °C.

The next day, the culture was centrifuged over three rounds of 2 mL each for 3 min at 6500 ×g, disposing of the supernatant. The pellet was fully resuspended by vortexing with 600 µL of MQ-H_2O. Then, 100 µL of cell lysis buffer were added and the reaction tube was carefully inverted for mixing. Afterward, 350 µL of neutralization solution were added, and the tube was inverted until a full color change to yellow occurred.

The solution was centrifuged for 10 min at 21 300 ×g. Afterward, the supernatant was pipetted onto a PureYield™ mini column. The column was centrifuged for 15 s at 21 300 ×g, and the flow-through was discarded.

200 µL of endotoxin removal wash were added to the column, followed by centrifugation for 15 s at 21 300 ×g, again discarding the flow-through. Then, 400 µL of column wash solution were added to the column, followed by centrifugation for 30 s at maximum speed. The flow-through was discarded.

The column was transferred to a new reaction tube. Elution was performed using 30 µL of nuclease-free water. After application of the water and incubation for 1 min at room temperature, the column was centrifuged for 15 s at 21 300 ×g. The concentration of the plasmid in the flow-through was determined using the NanoDrop™ One.

The purified plasmids were diluted to a final concentration of 100 ng µL^{-1}. In a Mix2Seq kit, 5 µL of the plasmid was mixed with 5 µL of pLenSeq fwd (10 µM) in one compartment and with 5 µL of pLenSeq rev (10 µM) in another compartment. The kit was sent for sequencing.

8.1.8 Plasmid Midi-Preparation

8.2 Protein Expression and Purification

8.2.1 Cell-Free Protein Expression

Cell-Free protein expression of the constructs d29-A, S-Tag-A, and S-Tag-B was conducted using an ALiCE® for Research kit. Before starting the reaction, the plasmids pLenEx-d29-A, pLenEx-S-Tag-A, pLenEx-S-Tag-B, pLenEx-PVX-CP (TODO Lookup name), and pLenEx-Strep-eYFP (all from purification using a NucleoBond Xtra Midi kit) were concentrated using speed vacuuming at 30 °C to concentrations between 1400 ng µL^{-1} and 3300 ng µL^{-1}. The volumes of plasmid DNA to be used for the cell-free reaction were calculated according to Equation 11. The final plasmid concentration was chosen as 50 nM.

$$V_{\mathrm{DNA}}\,[\mu\mathrm{L}] = \left(L_{\mathrm{plasmid}}\,[\mathrm{bp}] \cdot 618\ \frac{\mathrm{g}}{\mathrm{mol}\cdot\mathrm{bp}}\right) \cdot\ V_{\mathrm{reaction}}\,[\mu\mathrm{L}] \cdot \left(\frac{c_{\mathrm{final}}\,[\mathrm{nM}]}{c_{\mathrm{stock}}\,[\mathrm{ng}/\mu\mathrm{L}]}\right) \cdot\ 10^{-6} \tag{11}$$

Before starting the reaction, the 50 µL ALiCE tubes were fully thawed in a heating block at 25 °C. The solution was mixed by inverting the tubes and centrifuged for 5 seconds at 100 ×g to collect the liquid. After centrifugation, the tubes were placed on ice. The lids were perforated with a single hole using a needle (TODO Lookup diameter). The appropriate volume of the plasmids pLenEx-d29-A,

pLenEx-S-Tag-A, pLenEx-S-Tag-B, pLenEx-PVX-CP (TODO Lookup name), and pLenEx-Strep-eYFP were added to the respective reaction tubes. Additionally, a non-template control was set up by adding 2 µL MQ-H_2O. The reaction was incubated at 25 °C on an Eppendorf ThermoMixer at 700 rpm for 48 h. Afterward, the reaction tubes were placed on ice to stop the reaction, before being frozen at −20 °C.

8.2.2 Protein Purification Using Capto Core 700

Following cell-free protein expression, size exclusion chromatography using Capto™ Core 700 multimodal chromatography resin was conducted to purify large particles. 1 mL Capto™ Core matrix was suspended in 3 mL of 0.1 M phosphate buffer (pH 7.2) within a column and full sedimentation was awaited. 30 µL of the cell-free expression solution were applied to the column and incubated for 5 min. Afterward, the flow-through was collected. The column was cleaned using 3 mL of a solution out of 1.5 mL 30 % isopropyl alcohol and 1.5 mL 1 M NaOH. The column was stored in 20 % ethanol at 4 °C and reused multiple times. The chromatography was used on the samples d29-A, S-Tag-A, S-Tag-B, and PVX-CP.

8.3 Protein Analysis

8.3.1 SDS-PAGE

Samples from cell-free protein expression, both before and after purification with the Capto™ Core 700 column, were used in an discontinuous SDS polyacrylamide gel electrophoresis (SDS-PAGE) for further use with Coomassie Staining and Western Blot.

The composition of the resolving gel and the stacking gel are listed in Table **??**. All reagents used for the resolving gel, except for the ammonium persulfate (APS), were mixed together by vortexing. After addition of APS, the solution was shortly vortexed and about 5 mL of the gel were transferred into a 0.75 mm thick chamber for polymerization. Directly afterward, isopropyl alcohol was added to the top of the gel.

After polymerization, the isopropyl alcohol was removed using Whatman paper. The components for the stacking gel were mixed, APS was added, and the stacking gel was poured on top of the resolving gel. A comb was inserted into the stacking gel to create sample pockets.

Table 9: Composition of discontinuous SDS-PAGE gels. The amounts are shown for the preparation of two gels.

Component	**Resolving Gel (T = 12%)**	**Stacking Gel (T = 4%)**
MQ-H_2O	2.115 mL	3.645 mL
Tris-HCl stock (1 M)	3.75 mL (pH 8.8)	625 µL (pH 6.8)
AA stock (30%)	4 mL	830 µL
SDS (10%)	100 µL	50 µL
TEMED	10 µL	5 µL
APS (20%)	30 µL	15 µL
Total Volume	10 mL	5 mL

After polymerization of the stacking gel, the gel was either directly used in electrophoresis, or wrapped into wet paper and stored at 4 °C for up to a week.

For electrophoresis, the gel was placed vertically in a chamber containing SDS running buffer. The samples were mixed with 5x reducing loading buffer in a 4:1 ratio and boiled for 5 min. 10 µL sample volume was transferred into the gel pockets, and the marker Color Prestained Protein Standard, Broad Range (10-250 kDa) by New England Biolabs was used as marker. The gel was run at 180 V for about one hour.

8.3.2 Coomassie Staining

After completion of the SDS-PAGE, the gels were placed in Coomassie Staining solution for 30 minutes, while being gently swiveled on an orbital stainer. The Coomassie Staining solution was removed, and destaining solution was added to the gel, still being swiveled. The destaining solution was replaced multiple times, before destaining was complete.

8.3.3 Western Blot

For immunologic detection of specific epitopes on the SDS gel, Western Blotting was conducted. The gel was placed in semi-dry transfer buffer, and eight layers of Whatman paper as well as a nitrocellulose membrane were soaked in semi-dry transfer buffer for 5 min. Then, a stack of four layers of Whatman paper, the nitrocellulose membrane, the gel, and four layers of Whatman paper, was assembled in the blotting chamber of a Trans-Blot® Turbo™ machine. Transfer to the membrane took place at a constant voltage of 25 V and a current of maximally 1 A.

After blotting, the membrane was cut to the size of the gel, placed into 10 mL blocking buffer and incubated for 30 min under swiveling. The blocking buffer was removed, and the membrane was washed three times with PBS buffer, waiting 5 min between each exchange of the buffer. The primary antibody (either Rabbit-Anti-PVX in a 1:5000 ratio or Mouse-Anti-S-Tag in a 1:10000 ratio) was dissolved in 10 mL PBS buffer and added to the membrane. Incubation with the primary antibody was conducted overnight at room temperature or over the weekend at 4 °C. Afterward, the three washing steps were repeated and the secondary antibody was added (either Goat-Anti-Rabbit FC AP or Goat-Anti-Mouse FC AP, both in a 1:5000 ratio in PBS).

8.3.4 ELISA

8.3.5 Electron Microscopy

9 Results

9.1 DNA Cloning

9.2 Protein Expression

9.2.1 Coomassie Staining

9.2.2 Western Blot

9.2.3 ELISA

9.2.4 Electron Microscopy

Appendix A: Supplementary Figures

...

Appendix B: Algorithms

Algorithm 2 Sample Diffusion

def SampleDiffusion($\{\mathbf{f}^*\}$, $\{\mathbf{s}_i^{\text{inputs}}\}$, $\{\mathbf{s}_i^{\text{trunk}}\}$, $\{\mathbf{z}_{ij}^{\text{trunk}}\}$, Noise Schedule $[c_0, c_1, \ldots, c_T]$, $\gamma_0 = 0.8$, $\gamma_{\min} = 1.0$, noise scale $\lambda = 1.003$, step scale $\eta = 1.5$):

$\vec{\mathbf{x}}_l \sim c_0 \cdot \mathcal{N}(\vec{\mathbf{0}}, \mathbf{I}_3)$ $\quad \vec{\mathbf{x}}_l \in \mathbb{R}^3$

for all $c_\tau \in [c_1, \ldots, c_T]$ **do**

  $\{\vec{\mathbf{x}}_l\} \leftarrow$ CentreRandomAugmentation$(\{\vec{\mathbf{x}}_l\})$

  $\gamma \leftarrow \gamma_0$ if $c_\tau > \gamma_{\min}$ else 0

  $\hat{t} \leftarrow c_{\tau-1}(\gamma + 1)$

  $\vec{\boldsymbol{\xi}}_l \leftarrow \lambda\sqrt{\hat{t}^2 - c_{\tau-1}^2} \cdot \mathcal{N}(\vec{\mathbf{0}}, \mathbf{I}_3)$ $\quad \vec{\boldsymbol{\xi}}_l \in \mathbb{R}^3$

  $\vec{\mathbf{x}}_l^{\text{noisy}} \leftarrow \vec{\mathbf{x}}_l + \vec{\boldsymbol{\xi}}_l$

  $\{\vec{\mathbf{x}}_l^{\text{denoised}}\} \leftarrow$ DiffusionModule$(\{\vec{\mathbf{x}}_l^{\text{noisy}}\}, \hat{t}, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\})$

  $\vec{\boldsymbol{\delta}}_l \leftarrow (\vec{\mathbf{x}}_l - \vec{\mathbf{x}}_l^{\text{denoised}})/\hat{t}$

  $dt \leftarrow c_\tau - \hat{t}$

  $\vec{\mathbf{x}}_l \leftarrow \vec{\mathbf{x}}_l^{\text{noisy}} + \eta \cdot dt \cdot \vec{\boldsymbol{\delta}}_l$

end for

return $\{\vec{\mathbf{x}}_l\}$

Algorithm 3 Sample Diffusion with Symmetrization for Multimeric Complexes

def SampleDiffusion($\{\mathbf{f}^*\}$, $\{\mathbf{s}_i^{\text{inputs}}\}$, $\{\mathbf{s}_i^{\text{trunk}}\}$, $\{\mathbf{z}_{ij}^{\text{trunk}}\}$, Noise Schedule $[c_0, c_1, \ldots, c_T]$, $\gamma_0 = 0.8$, $\gamma_{\min} = 1.0$, noise scale $\lambda = 1.003$, step scale $\eta = 1.5$, Monomer Transforms $\{T_j\}$):

$\vec{\mathbf{x}}_l \sim c_0 \cdot \mathcal{N}(\vec{\mathbf{0}}, \mathbf{I}_3)$ $\quad\quad$ $\vec{\mathbf{x}}_l \in \mathbb{R}^3$

\# Modification: Initial Symmetrization

★: $(R_{\text{ref}}, \vec{\mathbf{t}}_{\text{ref}}) \leftarrow (\mathbf{I}, \text{mean}(\{\vec{\mathbf{x}}_l^{(1)}\}))$ $\quad\quad$ Denoted as $T_{\text{ref}} = (R_{\text{ref}}, \vec{\mathbf{t}}_{\text{ref}})$

★: $\vec{\mathbf{x}}_l^{(j)} \leftarrow T_{\text{ref}} \circ T_j \circ T_{\text{ref}}^{-1} \circ \vec{\mathbf{x}}_l^{(1)}$

for all $c_\tau \in [c_1, \ldots, c_T]$ **do**

\# Track Origin of Symmetrization Center

★: $\vec{\mathbf{t}}_{\text{ref}} \leftarrow \text{mean}(\{\vec{\mathbf{x}}_l\}_{l \in I_1})$

$\{\vec{\mathbf{x}}_l\}, T_{\text{aug}} \leftarrow \text{CentreRandomAugmentation}(\{\vec{\mathbf{x}}_l\})$

\# Track Movement by CentreRandomAugmentation

★: $T_{\text{ref}} \leftarrow T_{\text{aug}} \circ T_{\text{ref}}$

$\gamma \leftarrow \gamma_0$ if $c_\tau > \gamma_{\min}$ else 0

$\hat{t} \leftarrow c_{\tau-1}(\gamma + 1)$

$\vec{\boldsymbol{\xi}}_l \leftarrow \lambda\sqrt{\hat{t}^2 - c_{\tau-1}^2} \cdot \mathcal{N}(\vec{\mathbf{0}}, \mathbf{I}_3)$ $\quad\quad$ $\vec{\boldsymbol{\xi}}_l \in \mathbb{R}^3$

$\vec{\mathbf{x}}_l^{\text{noisy}} \leftarrow \vec{\mathbf{x}}_l + \vec{\boldsymbol{\xi}}_l$

$\{\vec{\mathbf{x}}_l^{\text{denoised}}\} \leftarrow \text{DiffusionModule}(\{\vec{\mathbf{x}}_l^{\text{noisy}}\}, \hat{t}, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\})$

\# Recenter and Symmetrize Denoised Prediction

★: $\vec{\mathbf{x}}_l^{\text{denoised}} \mathrel{+}= \text{mean}(\{\vec{\mathbf{x}}_l^{(1),\text{noisy}}\}) - \text{mean}(\{\vec{\mathbf{x}}_l^{(1),\text{denoised}}\})$

★: $\vec{\mathbf{x}}_l^{(j),\text{denoised}} \leftarrow T_{\text{ref}} \circ T_j \circ T_{\text{ref}}^{-1} \circ \vec{\mathbf{x}}_l^{(1),\text{denoised}}$

$\vec{\boldsymbol{\delta}}_l \leftarrow (\vec{\mathbf{x}}_l - \vec{\mathbf{x}}_l^{\text{denoised}})/\hat{t}$

$dt \leftarrow c_\tau - \hat{t}$

$\vec{\mathbf{x}}_l \leftarrow \vec{\mathbf{x}}_l^{\text{noisy}} + \eta \cdot dt \cdot \vec{\boldsymbol{\delta}}_l$

end for

return $\{\vec{\mathbf{x}}_l\}$

Appendix C: Synthetic Sequences

C.1: Construct A - Protein Sequence (Designed)

ASGLFTIPDGDFFDTVRHIVASNAVATNEDLSKIEALWKDMKVPTDTLFQAAVDLCRHCA
DVGSSAQTEMIGTGPYSNGISFARLAAAIRQVCTLRQFCMKYAPVVWNWMLTNNSPPANW
QARGFKPEHKFAAFDFFDGVTNPAAIMPKEGLLRPPSEAEMIAAHTAAEVKSTKARAQSN
DFASLDAAVTRGRITGQTTAEAVVTIPPP

C.2: Construct B - Protein Sequence (Designed)

ATGLNTVPDGDYFKTVKHKVLSNRVATDAELAAIETKWLAAGVPAATLFQAALDLCFQAA
DIGCGEDTVFVGTGPYTNGVSFQDLAAAIIRQVTTLLKFCRRYAPCVWNYMLTHNLPPADW
LARGFYPDHRYAAFDFFDGVENPAAIQPKLGLLRPPTVAERIAYHTLKTITTTTAAAAGN
DFASLHTAVTRGRLTGQSAAERIIHIPAA

C.3: Construct A - DNA Sequence (d29-A)

GCTTCTGGCTTATTCACCATACCTGACGGGGATTTCTTCGATACTGTAAGGCACATTGTA
GCTAGTAATGCTGTGGCAACAAACGAGGATCTCAGCAAGATCGAGGCTTTGTGGAAAGAC
ATGAAAGTTCCTACTGACACTCTTTTCCAGGCTGCCGTCGATCTGTGCCGACATTGTGCA
GACGTTGGGAGCTCTGCTCAGACAGAAATGATCGGTACTGGACCATATTCAAACGGAATA
TCATTTGCAAGACTGGCCGCTGCCATCCGACAAGTATGCACTTTGCGACAATTTTGTATG
AAGTACGCACCTGTTGTCTGGAATTGGATGTTGACTAATAATTCTCCACCCGCAAACTGG
CAGGCCAGAGGCTTCAAGCCAGAACACAAGTTTGCTGCATTTGACTTCTTCGACGGAGTT
ACTAATCCTGCCGCAATCATGCCTAAAGAAGGATTGTTACGACCCCCATCCGAGGCCGAG
ATGATCGCAGCTCATACTGCAGCCGAGGTCAAGAGCACAAAAGCTCGAGCACAAAGCAAT
GACTTCGCTTCCCTTGACGCAGCCGTCACACGAGGGCGAATCACCGGCCAGACTACAGCA
GAGGCCGTTGTGACCATTCCCCCCCCT

C.4: Construct B - DNA Sequence (d29-B)

GCCACCGGGCTGAATACCGTCCCTGACGGCGACTATTTCAAAACAGTCAAGCACAAGGTT
TTATCCAATAGAGTAGCAACTGATGCTGAGTTGGCCGCTATTGAGACAAAATGGCTGGCT
GCAGGGGTACCAGCAGCAACACTCTTCCAAGCCGCTCTGGACCTTTGTTTTCAGGCTGCC
GACATCGGTTGTGGTGAGGACACAGTATTCGTCGGGACCGGACCTTACACCAATGGCGTT
AGCTTCCAAGACCTCGCTGCCATCATAAGGCAAGTGACTACCTTATTGAAGTTCTGCCGA
CGTTACGCCCCCTGTGTATGGAACTATATGTTGACCCACAATCTTCCACCCGCAGATTGG
CTGGCACGTGGCTTCTATCCTGACCACAGGTATGCTGCCTTCGATTTCTTCGACGGCGTA
GAGAATCCTGCTGCTATCCAACCAAAATTGGGTCTGCTTCGTCCCCCTACTGTTGCTGAG
AGGATCGCTTACCATACCCTGAAGACCATTACAACAACCACCGCAGCCGCCGCAGGTAAC
GATTTTGCTTCACTTCATACTGCTGTAACTCGTGGTAGACTCACAGGTCAGAGCGCCGCT
GAGAGAATCATACACATACCCGCAGCT

C.5: Construct A - DNA Sequence with S-Tag (S-Tag-A)

ATGAGTAAAGAAACAGCCGCCGCTAAATTCGAACGTCAGCATATGGATAGTCCTGCATCA
ACAACCCAACCCATAGGTAGCACCACTAGCACAACTACTAAGACTGCCGGTGCAACCCCT
GCTACCGCTTCTGGCTTATTCACCATACCTGACGGGGATTTCTTCGATACTGTAAGGCAC

ATTGTAGCTAGTAATGCTGTGGCAACAAACGAGGATCTCAGCAAGATCGAGGCTTTGTGG
AAAGACATGAAAGTTCCTACTGACACTCTTTTCCAGGCTGCCGTCGATCTGTGCCGACAT
TGTGCAGACGTTGGGAGCTCTGCTCAGACAGAAATGATCGGTACTGGACCATATTCAAAC
GGAATATCATTTGCAAGACTGGCCGCTGCCATCCGACAAGTATGCACTTTGCGACAATTT
TGTATGAAGTACGCACCTGTTGTCTGGAATTGGATGTTGACTAATAATTCTCCACCCGCA
AACTGGCAGGCCAGAGGCTTCAAGCCAGAACACAAGTTTGCTGCATTTGACTTCTTCGAC
GGAGTTACTAATCCTGCCGCAATCATGCCTAAAGAAGGATTGTTACGACCCCCATCCGAG
GCCGAGATGATCGCAGCTCATACTGCAGCCGAGGTCAAGAGCACAAAAGCTCGAGCACAA
AGCAATGACTTCGCTTCCCTTGACGCAGCCGTCACACGAGGGCGAATCACCGGCCAGACT
ACAGCAGAGGCCGTTGTGACCATTCCCCCCCCTTAA

C.6: Construct B - DNA Sequence with S-Tag (S-Tag-B)

ATGAGCAAGGAGACTGCTGCAGCCAAGTTTGAGCGTCAGCACATGGACAGCCCAGCTTCA
ACCACTCAACCCATCGGTTCTACTACATCCACAACTACCAAGACAGCAGGGGCTACTCCA
GCCACAGCCACCGGGCTGAATACCGTCCCTGACGGCGACTATTTCAAAACAGTCAAGCAC
AAGGTTTTATCCAATAGAGTAGCAACTGATGCTGAGTTGGCCGCTATTGAGACAAAATGG
CTGGCTGCAGGGGTACCAGCAGCAACACTCTTCCAAGCCGCTCTGGACCTTTGTTTTCAG
GCTGCCGACATCGGTTGTGGTGAGGACACAGTATTCGTCGGGACCGGACCTTACACCAAT
GGCGTTAGCTTCCAAGACCTCGCTGCCATCATAAGGCAAGTGACTACCTTATTGAAGTTC
TGCCGACGTTACGCCCCCTGTGTATGGAACTATATGTTGACCCACAATCTTCCACCCGCA
GATTGGCTGGCACGTGGCTTCTATCCTGACCACAGGTATGCTGCCTTCGATTTCTTCGAC
GGCGTAGAGAATCCTGCTGCTATCCAACCAAAATTGGGTCTGCTTCGTCCCCCTACTGTT
GCTGAGAGGATCGCTTACCATACCCTGAAGACCATTACAACAACCACCGCAGCCGCCGCA
GGTAACGATTTTGCTTCACTTCATACTGCTGTAACTCGTGGTAGACTCACAGGTCAGAGC
GCCGCTGAGAGAATCATACACATACCCGCAGCTTAA

References

[1] Josh Abramson et al. "Accurate structure prediction of biomolecular interactions with AlphaFold 3". In: *Nature* 630.8016 (2024), pp. 493–500. DOI: 10.1038/s41586-024-07487-w. URL: https://doi.org/10.1038/s41586-024-07487-w.

[2] Nathaniel R. Bennett et al. "Improving de novo protein binder design with deep learning". In: *Nature Communications* 14.1 (2023), p. 2625. DOI: 10.1038/s41467-023-38328-5. URL: https://doi.org/10.1038/s41467-023-38328-5.

[3] J. Dauparas et al. "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615 (2022), pp. 49–56. DOI: 10.1126/science.add2187. eprint: https://www.science.org/doi/pdf/10.1126/science.add2187. URL: https://www.science.org/doi/abs/10.1126/science.add2187.

[4] Alessandro Grinzato et al. "Atomic structure of potato virus X, the prototype of the Alphaflexiviridae family". In: *Nature Chemical Biology* 16.5 (2020), pp. 564–569. ISSN: 1552-4469. DOI: 10.1038/s41589-020-0502-4. URL: https://doi.org/10.1038/s41589-020-0502-4.

[5] Jim Lawrence, Javier Bernal, and Christoph Witzgall. "A Purely Algebraic Justification of the Kabsch-Umeyama Algorithm". In: *Journal of Research of the National Institute of Standards and Technology* 124 (Oct. 2019). ISSN: 2165-7254. DOI: 10.6028/jres.124.028. URL: http://dx.doi.org/10.6028/jres.124.028.

[6] Elaine C. Meng et al. "UCSF ChimeraX: Tools for structure building and analysis". In: *Protein Science* 32.11 (2023), e4792. DOI: https://doi.org/10.1002/pro.4792. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pro.4792. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.4792.

[7] Soheil Sarabandi and Federico Thomas. "Solution methods to the nearest rotation matrix problem in : A comparative survey". In: *Numerical Linear Algebra with Applications* 30.5 (2023), e2492. DOI: https://doi.org/10.1002/nla.2492. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nla.2492. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nla.2492.

[8] Joseph L. Watson et al. "De novo design of protein structure and function with RFdiffusion". In: *Nature* 620.7976 (2023), pp. 1089–1100. DOI: 10.1038/s41586-023-06415-8. URL: https://doi.org/10.1038/s41586-023-06415-8.