RamaNet: Computational *De Novo* Protein Design using a Long Short-Term Memory Generative Adversarial Neural Network

Sari Sabban*[+1 2], Mikhail Markovsky[+3]

[1]Department of Biological Sciences, Faculty of Science, King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia

[2]Electrical and Computer Engineering Department, College of Engineering, Effat University, Jeddah, Kingdom of Saudi Arabia

[3]North Caucasian Federal Scientific Center of Horticulture, Viticulture, Wine-making, Krasnodar, Russian Federation

[+]These authors contributed equally to this work

*Correspondence to: Sari Sabban at sari.sabban@gmail.com

Abbreviations:
LSTM: Long Short-Term Memory
PDB: Protein Data Bank
PS: Phi (Φ) angle and Psi (Ψ) angle featured dataset
GAN: Generative Adversarial Network
MDN: Mixture Density Network
MLP: Multi Layer Perceptron
Abinitio and AbinitioRelax are used interchangeably to refer to the same folding simulation protocol

Keywords:
Protein Structure, Machine Learning, PDB, Neural Network, *De novo* Protein Design, LSTM, GAN

Abstract:
The ability to perform *De novo* protein design will allow researchers to expand the pool and variety of available proteins, by designing synthetic structures computationally they can make available more structures than are available in the Protein Data Bank, design structures that are not found in nature, or direct the design of proteins to acquire a specific desired structure. While some researchers attempt to design proteins from first physical and thermodynamic principals, we decided to attempt to perform *de novo* protein design statistically using machine learning by building a model that uses a long short-term memory generative adversarial neural network architecture. The LSTM based GAN model used the Φ and Ψ angles of each residue from an augmented dataset of only helical protein structures. Though the network's output structures were not perfect, they were idealised and evaluated post prediction where the bad structures were filtered out and the adequate structures kept. The results were successful in developing a logical, rigid, compact, helical protein backbone topology. This backbone topology was then used to computationally design side chains that should allow the final protein to fold to the designed structure.

Introduction:

The concept of amino acid sequences folding into globular protein molecules allows for proteins' large functional diversity, making them mediate all the functional aspects of living organisms, thus winning themselves attention from biochemists for decades. Everything in this universe can be described by a mathematical algorithm, including living organisms and their functional units. The fusion of machine learning with computational biology is accelerating research in both fields and bringing humanity closer to the setup of performing most biological research quickly, cheaply, and safely *in silico* while only translating the very crucial aspects of it. Having access to a large database of protein crystal structures would naturally result in the use of machine learning to design proteins computationally.

De novo protein design (i.e from the beginning) is very well explained in this review[1]. Proteins fold into a specific shape depending on the sequence of their amino acids, and of course shape dictates function. The driving forces that allows proteins to fold are the hydrogen bond interactions within the backbone and between the side chains, the Van der Waals forces, and principally the interaction of hydrophobic side chains within the core. The space of all possible sequences for all protein sizes is extremely large (as an example there are 20^{200} possibilities for a 200 residue protein). Thus is it not surprising that natural proteins exist in clusters close to each other, which is logical since proteins would evolve away from a central functional protein to fold correctly and acquire new folds and functions rather than go through the tedious ordeal of finding a totally new protein structure within the space of all possibilities. Thus, even though the protein data bank adds about 10,000 new structures to its repository every year, most of these new structures are not unique folds.

The relationship between the sequence of a protein and its specific structure is understood, but we still lack a unified absolute solution to calculate one from the other. Hence why current man-made protein designs are usually the result of evolving already natural proteins[2] since randomly finding a functionally folded protein from the space of all possible protein sequences is more or less impossible.

Current solutions employ an energy function, such as REF15, that uses some fundamental physical theories, statistical mechanical models, and observations of protein structures in-order to approximate the potential energy of a protein[3]. Knowing the protein energy allows us to guide our search for the structure of a protein given its sequence (the structure resides at the global energy minima of that protein) thus attempting to connecting the sequence of a protein with its structure. The critical item is in the energy function, the higher its accuracy the higher our confidence in knowing the computer structure is the real natural structure. Thus using the energy function to perform Structure prediction (going from a known sequence to find the unknown three dimensional structure) can also be used to perform fixed-backbone design (going from a known three dimensional structure to find the sequence that folds it). Where as *de novo* design (neither backbone nor sequence is known), knowing one results in finding the other using the same energy function[1].

Some researchers attempted *de novo* protein design by designing a topology from assembling short sequence peptide fragments taken from natural protein crystal structures[4] [5], these fragments are calculated statistically depending on the secondary structures they are found in. Sometimes these fragment system is combined with first physical principals to model the loops between secondary structures to achieve a desired three dimensional topology[6]. Others have used parametric equations to study and specify the desired protein geometry[7] [8] [9] [10] [11] [12].

On the other hand, some researchers have used machine learning for protein design, using the constraints (Cα-Cα distances) as the input features of the network and using a sliding window to read a sequence of residues, getting their types and constraints, then predicting the next one, giving the output prediction as an amino acid sequence[13], reporting an accuracy of 38.3%.

The deep neural network architecture that we chose is the Long Short-Term Memory (LSTM) based Generative Adversarial Network (GAN) architecture[37]. The LSTM is usually used in natural language and data sequence processing, but in our model the LSTM was setup in a slightly different manner.

The model is made up of two networks that work against each other, the first is a generator network that is made up of a stack of LSTM layers, followed by fully connected layers, followed by a Mixture Density Network (MDN) and works by using random noise numbers as input to build the values for the Φ and Ψ angles. The other network is a discriminator that is made up of a stack of LSTM layers followed by fully connected layers and works to studying the dataset and determining whether the output from the generator is a truly logical structure or not[30].

Our effort in this paper is to use machine learning to learn the general fold of natural proteins, and using this generalising statistical concept we can design a novel yet logical protein backbone topology, thus getting the three dimensional structure, and knowing that we can use the REF15 energy function to perform fixed-backbone design to get the sequence that will fold the protein into that structure. This is not by any means a guaranteed endeavour, the available energy functions are not yet perfect, thus we cannot perform fixed-backbone design and be confident that our computed structure will definitely fold correctly in nature, but the tools we have can at least guide us towards the correct sequence. Our research at this moment is only concerned with getting a new and unique folded ideal helical protein and not it found in nature rather than a protein with a specific function or a specific structure, thus our system results in random yet compact helical structures.

Method:

The following steps were used to generate a dataset with the required features, setup and train the network, optimise the output, then evaluate the results.

Data generation:

The entire PDB database was downloaded (~150,000 structures) and each entry was divided into its constituent chains to become individual separate structures (i.e: each structure is made of only a single chain). Then each structure was analysed and chosen if it contained the following criteria: made up of only polypeptides, size between 80 and 150 amino acids, no breaks in the chain (must be a continuous polypeptide), the sum of residues that make helices and sheets must be larger than the sum of amino acids making up loops, and the final structure must have an Rg (radius of gyration) value less than 15. the remaining structures were then filtered by a human eye to make sure only the desired structure concepts are kept (i.e: keeping rigid compact structures and removing loose open structures that slipped through the filtering process), also trying to keep a diversity of structure folds rather than numerous repeats of the same fold (the haemoglobin fold was quit abundant). In previous attempts we tried to use a mixture of structures (some structures were only helices, some were only sheets, and some were a mix of the two) but that proved challenging in optimising the network, as such we settled

for a dataset made up of only helical protein structures. The final database had 607 ideal helical structures. These structures were then cleaned (non-amino acid atoms were removed) in preparation to push them through the Rosetta modelling software that only takes in polypeptide molecules.

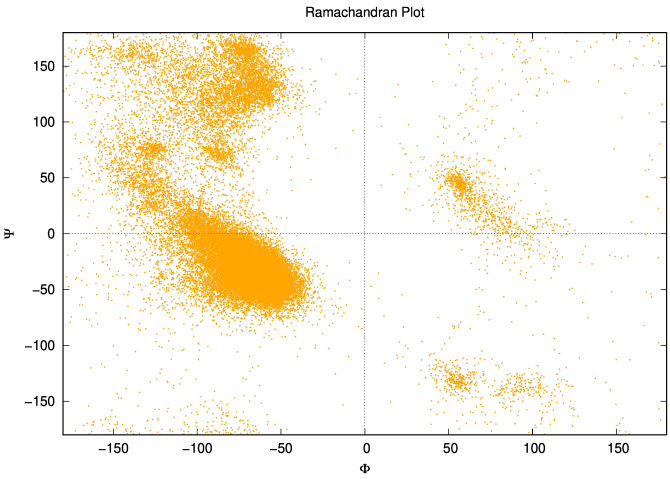


Figure 1: Ramachandran plot of the dataset showing the Φ and Ψ angles of each amino acid for each structure. This is the un-augmented data of structures that are only made of helices. One point to note, the angles here are represented between the range -180° to 180° as is conventional, while in the actual dataset the range is from 0° to 360°.

Data augmentation:
These 607 structures were augmented using the Rosetta FastRelax protocol[14]. This protocol performs multiple cycles of packing and minimisation. In other words, it performs small slight random moved on the backbone and side chains. Its original intended function is to move the structure slightly to find the conformation of the backbone and side chains that corresponds to the lowest energy state. Since the protocol performs random moves, a structure relaxed on two separate occasions will result in two distinct molecules that look exactly the same with similar minimum energy scores, but technically have different Φ and Ψ angle values. This is the concept we used to augment our structures, and each structure was relaxed 500 times to give a final database of 303,500 structures.

Feature extraction:
We found that using only the Φ and Ψ angles values from a crystal structure we were able to fold a structure to its correct native fold, thus we settled for only using the Φ and Ψ angles of the structures as the only relevant features required to correctly fold a structure, (figure 1) details the range of angles in the un-augmented data. Each amino acid's Φ and Ψ angel values were extracted and put in a table as in (table 1). This is the dataset we used to train the neural network.

Table 1: The first 5 examples of the PS_Helix_500.csv dataset showing the PDB ID_chain_augmentation number, residue 1 Φ angle, residue 1 Ψ angle, all the way to residue 150. 360.0° was used for the first missing angle while 0.0° was used to represent no residue.

	PDB_ID	phi_1	psi_1	phi_2	psi_2	phi_3	psi_3	.....	phi_150	psi_150
1	1TQG_A_0293.pdb	360	98.8	207.4	163.8	298.1	313.6	.....	0.0	0.0
2	1EZ3_A_0261.pdb	360	277.8	208.3	37	306.7	316.4	.....	0.0	0.0
3	5IP0_E_0241.pdb	360	86.7	293.2	328.7	292.2	313.1	.....	0.0	0.0
4	2P5T_G_0123.pdb	360	185.9	254.3	176.2	308.4	139.3	.....	0.0	0.0
5	5EOH_A_0211.pdb	360	144.4	293.5	334.6	294.7	320.9	.....	0.0	0.0

The neural network:
The model (figure 2) was built using the SenseGen model as a basis[30] and consisted of two networks: a generator *G* network and a discriminator *D* network. The *G* network was constructed from an LSTM layer with 64 nodes, followed by two dense fully connected MLPs with 32 nodes for the first and 12 nodes for the second and both employing a sigmoid activation, then followed by an MDN layer. The training was done using the Adam optimiser through time with a loss function defined to increase the likelihood of generating the next time step value.

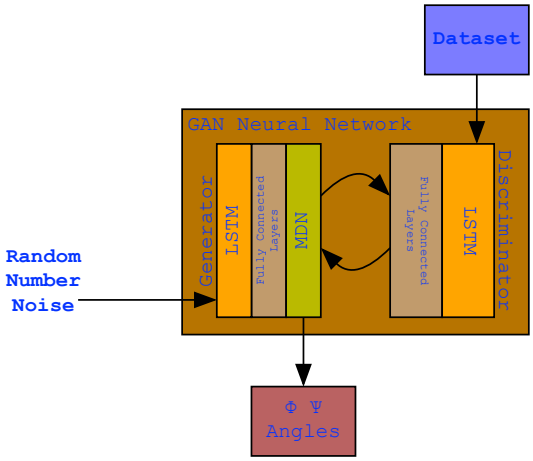


Figure 2: The LSTM based GAN model employing a generative network and a discriminator network that work adversarially against each other.

The *D* network was constructed from an LSTM layer with 64 nodes, followed by a dense fully connected MLP layer with 32 nodes followed by a single dense MLP unit layer employing a sigmoid activation function, so the output of this network is a prediction; the probability of the data as real (1) or fake (0). The network employed the cross-entropy loss function.

The *G* network used random noise as a starting seed, this noise was generated by taking a single randomly distributed number [0, 1) as the first predicted value, the number was then reshaped to the last item of the predicted value to (batch_size, num_steps, 1). The network predicts the main parameters of the new value (μ, σ, π) several times (according to the num_mixtures value) and selects the single mixture randomly but according to the π value. It then predicts the next value according to the normal distribution using the μ and σ values. It adds the final value to the prediction chain and returns to the step 2 until the predefined sequence length is obtained. The initial random number was stripped from the returned sequence.

Once the networks were constructed, the dataset was normalised and the training was done as follows for each adversarial epoch:

1. Sample minibatch from dataset (X_{true}).
2. Sample minibatch from *G* network (X_G).
3. Train the *D* network on the training set (X_{true}, X_G).
4. Sample minibatch from dataset (X_{true}).
5. Sample minibatch from *G* network (X_G).
6. Train the *G* network on the training set (X_{true}).

Post backbone topology generation processing and filtering:
The output of the neural network is always a structure with 150 amino acids, thus the ends were trimmed if they have no helical structures. The Φ and Ψ angel values are used to fold the protein, but since the majority of the structures end up with helical secondary structures but still with an open conformation, the structure is relaxed using the Rosetta FastRelax protocol to idealise the helices and compact the structure. Furthermore, not every prediction from the neural network results in an ideal structure even after the relax step, therefore we employed a filter to filter out bad structures. The filter discards structures that are less than 80 amino acids, has more residues in loops than in helices, has less than 15% residues making up its core, and has a maximum distance between Cα1 and any other Cα greater than 88 Å (the largest value in the dataset).

Flexible-backbone design:
The neural network's job is to give a backbone topology, this backbone topology is constructed out of valines. The next step is to design the structure to reach the best amino acid sequence that will result in the protein folding into its intended designed backbone. This is done by flexible-backbone design, based on RosettaDesign[15 33 34] using the Rosetta modelling software. Technically PyRosetta was used[16] since it was easier to integrated the code with the neural network's python script and combine the Rosetta engine with Biopython[31] and DSSP[32 34] to tailor each residue's amino acids during flexible-backbone design. The flexible-backbone design protocol starts with generating a blueprint file of the different residues and their corresponding secondary structures highlighting the loop positions to be rebuilt while keeping the secondary structures fixed, then a resfile (residue file) that lists the allowed amino acids to be used at each residue position depending on the residue's SASA (solvent accessible surface area) and secondary structure[17] (table 2).

Table 2: The allowable amino acids at each residue position relative to the secondary structure and position within the layers of the structure[17]. Surface amino acids of helical and sheet secondary structures were defined by a SASA value $\geqslant$ 60 Å while for loops it was $\geqslant$ 40 Å. Boundary amino acids of helical and sheet secondary structures were defined by a SASA value between 15 Å and 60 Å while for loops it was between 25 Å and 40 Å. Core amino acids of helical and sheet secondary structures were defined by a SASA value $\leqslant$ 15 Å while for loops it was $\leqslant$ 25 Å. Though the network was trained to generate only helical structures these rules were still implemented.

	Surface	Boundary	Core
Helix	QEKH	AVILWQEKFM	AVILFWM
Sheet	QTY	AVILFYWQTM	AVILFWM
Loop	PGNQSTDERKH	AVILFYWGNQSTPDEKR	AVILPFWM

The protocol then starts with relaxing[14] the structure to find the energy minima using the FastRelax mover, then the loops are remodelled using the BluePrintBDR[34] mover which replaces the loops with fragments, then comes the flexible backbone design[5 15 33] that allows for side chain as well as backbone movement using the FastDesign mover, then the structure is idealised to optimise the bond lengths and angles using the Idealise mover, then finally relaxed[14] one more time. The protocol was repeated 10 times by simulated annealing to result is a low energy decoy, 10 decoys were generated and the lowest energy decoy was chosen.

Structure prediction:
The proteins were computationally analysed for secondary structures using PSIPRED[29], and for crystallisability using XtalPred-RF[35].

The folding of the designed structures were computationally predicted initially using Rosetta AbinitioRelax structure prediction simulation[18]. The designed structure's FASTA sequence was submitted to the Old Robetta server that would generate fragments for the structure[19]. These fragments are randomly inserted into the primary structure of the

sequence and thus are used to direct the Rosetta AbinitioRelax protocol into finding the lowest energy conformation for the designed structure, in other words predict the possible fold of the designed amino acid sequence.

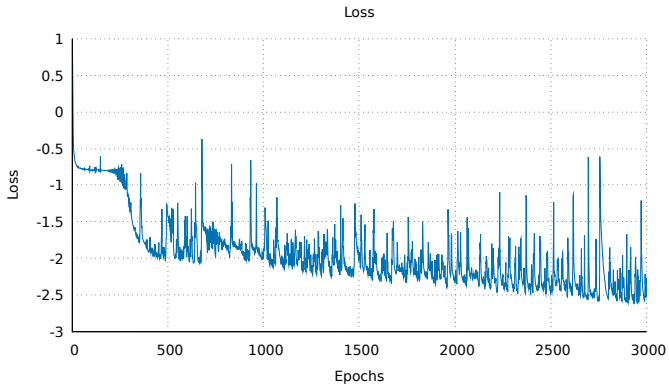


Figure 3: The mean loss of the whole network over epoch, for 3,000 epochs, showing a general downward trend, this indicates that subsequent epochs the *G* network gets better at generating structures that the *D* network correctly classifies as real logical structures.

Since the Rosetta AbinitioRelax structure prediction simulation relies heavily on the fragments generated from natural protein folds (which is generated from a statistical analysis of the probability of a sequence corresponding to a secondary structure) we were not confident that this protocol would accurately find the lowest energy structure of any of the unnatural designed structure folds. Thus even though we employed the simulation the negative results were not surprising. But it still begs the question: which was suboptimal, the design protocol or the folding simulation? since they both use the same REF15 energy function.

We further attempted to computationally predict the structures of our designed proteins using the New Robetta Server[20 21] which uses the Rosetta AbinitioRelax protocol with added optimisations, Swiss-Model[22 23 24 25 26], RaptorX[27], and Phyre2[28].

The only definitive way to solve the real structure of our designed proteins was to either crystallise them, or analyse them through NMR, both of which are within the future plans.

Results:

We dubbed the dataset PS_Helix_500 due to the fact that the features used are the Φ and Ψ angles, only strictly helical protein structures were used, and each structure was augmented 500 times.

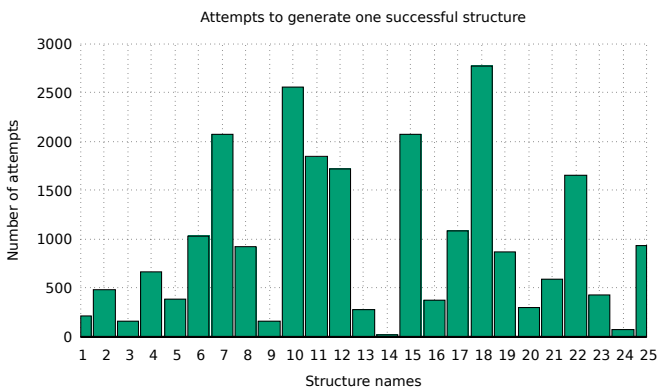


Figure 4: The number is attempts before a structure with desired structural characteristics is reached. The figure shows great variability ranging from 24 attempts to 2,778 attempts over the span of 25 different structures, giving an average success rate of ~0.40%.

The neural network was trained on the dataset for 3000 epochs with a generally sloping down mean loss (figure 3) indicating that the *G* gets better at generating data that the *D* classifies as real. The network was then used to generate the Φ and Ψ angles for a structure. Using these angles a primary straight structure with a length of 150 valine residues is constructed then the Φ and Ψ angles are applied to each residue. The resulting structure clearly shows the helical secondary structures, but the loops are usually not ideal, thus the structure does not come together into a compact topology. To correct this, the structure is relaxed and the structure brought together resulting in a structure's backbone topology with valines as a temporary sequence (figure 7) shows the difference between the relaxed and unrelaxed

angles of the structures. This setup is also not perfect at achieving an ideal structure every time, so a filter was deployed to filter out suboptimal structures by choosing a structure that have more residues with secondary structure helices than loops, a structure that is not smaller than 80 residues, and has more than 15% residues in its core. Generating multiple structures revealed that the success rate of the network is ~0.40% (figure 4) which can take between 20 minutes and 12 hours to generate a structure with the desired characteristics. Once a structure's backbone topology is generated the structure is sequence designed using flexible-backbone design to achieve a sequence that folds the protein to the designed structure. The best (low energy score) of 10 designed structures was chosen and its secondary structure predicted from its FASTA sequence using PSIPRED, and its predicted crystallisablity calculated using XtalPred-RF. The protocol is summarised in (figure 5), and the results are compiled in (figure 6).

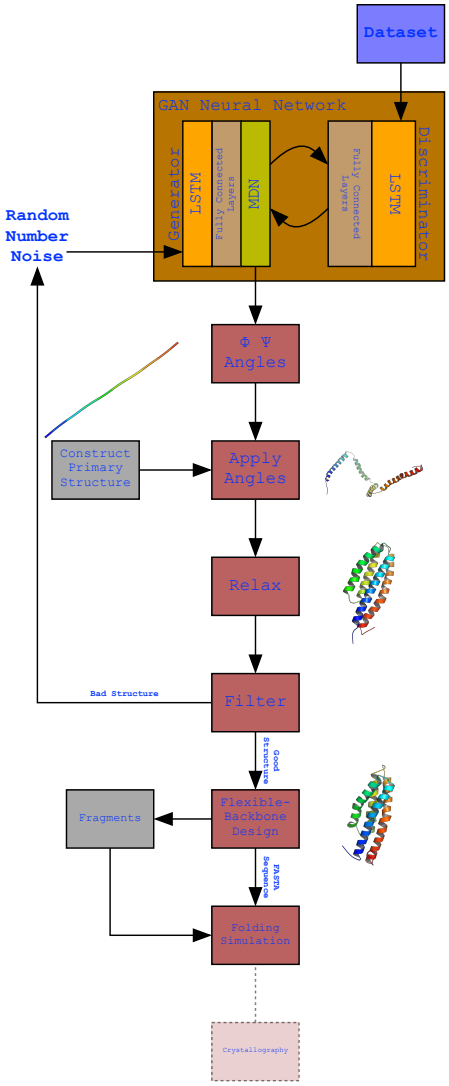
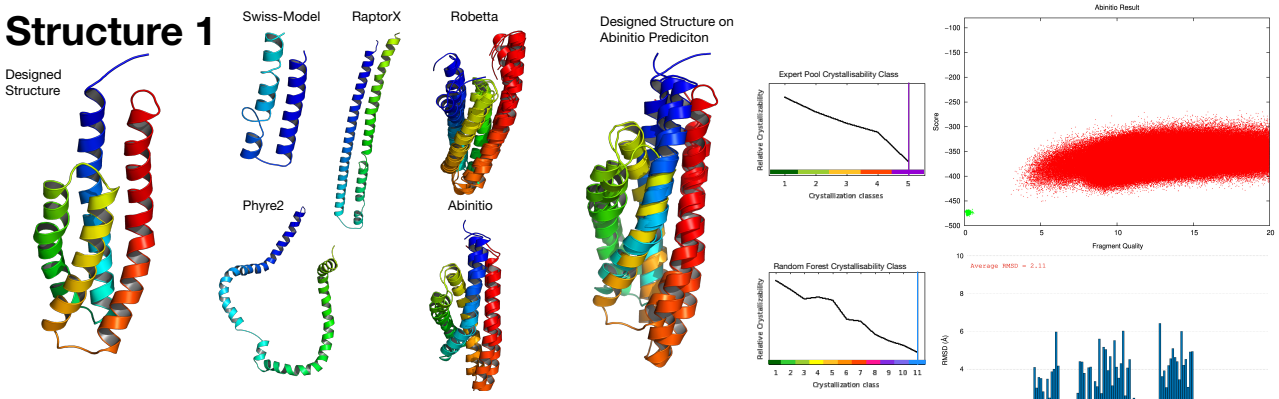


Figure 5: The full protocol showing the structure of the model and its output. The networks output are Φ and Ψ angles which are applied to a primary structure, which results in the development of the secondary structures but not a final compact structure due to suboptimal loop structures due to their variability in the dataset. To overcome this, the structure is relaxed to bring the secondary structure helices together. This does result in more compact structures but is not always ideal, thus a filter is used to filter our non-ideal structures and keep an ideal structure when generated. Once an idea structure is generated and passes through the filter it is flexible-backbone relaxed to find the ideal sequence that will fold the structure into the designed topology. The best lowest energy scoring decoy is chosen and its fold simulated using different folding algorithms.

The final designed structure's fold was then computationally simulated using different algorithms. We first generated fragments for the structure using the Robetta server and analysed their quality (the fragment with the lowest RMSD at each position of the structure) having the best RMSD values gives us a quick indication of how the *Abinitio* simulation will go since it will not find structures with lower RMSD than the lowest structure (figure 6). Using these fragments an *Abinitio* fold simulation was performed where the structure is folded using its FASTA sequence and assistance from the fragments and the PSIPRED secondary structure prediction. A plot of score against RMSD is plotted, a funnel shaped plot would indicate a successful simulation where the algorithm converges onto a local minima that has a high similarity to the designed structure (figure 6).

We further attempted to computationally predict the structures of our designed proteins using Robetta server which also uses the Rosetta AbinitioRelax protocol with added optimisations, Swiss-Model, RaptorX, and Phyre2, all the results are summarised in (figure 6). Since no algorithm is yet ideal to definitively fold a structure computationally, we were hoping to see if there is clear similarly between different algorithms since logical and simple ideal structure do end up with similar structures when their is simulated using these different algorithms.

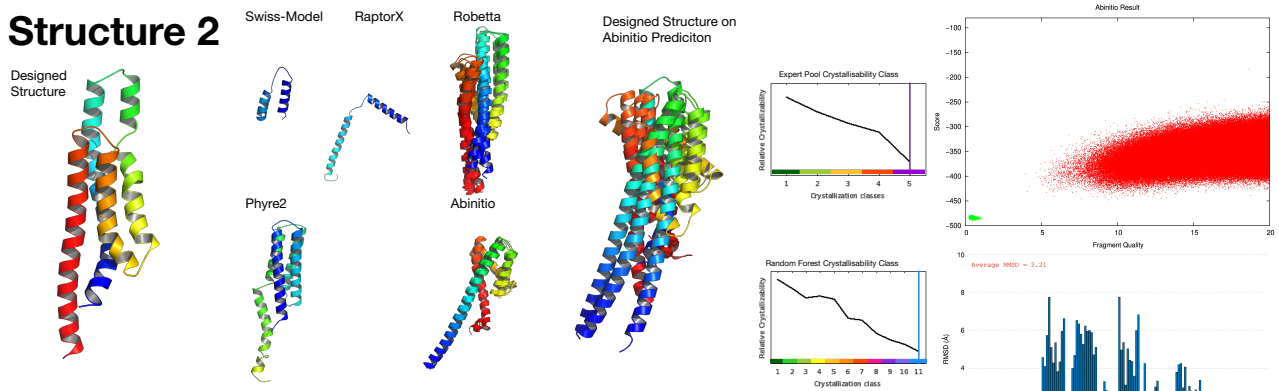
Structure 1



PSIPRED secondary structure prediction

```
Conf: 998515799999999999999999999999999983033338999999999999999999999999989960899999999999999999999999875457512999999999999999999999999999999999913656689
Pred: CCCCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCCCHHHHHHHHHHHHHHHHHHHHHHHHCCCHHHHHHHHHHHHHHHHHHHHHHHCCCHHHCHHHCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCCCCC
  AA: KGGGGEDAKKLRQEAAKKALKVVQEMLKEVAKAAAHKLDKGEEAEKVAKKVLKQMQKEVAEKAMKEVGPDGAAQKIVEKAAKEAIKKLIHQAAKEIGTENATKGKEAQEEAEKEAKKVFKEKAKKLAEQLKKKHEAAEKGETATDD
```

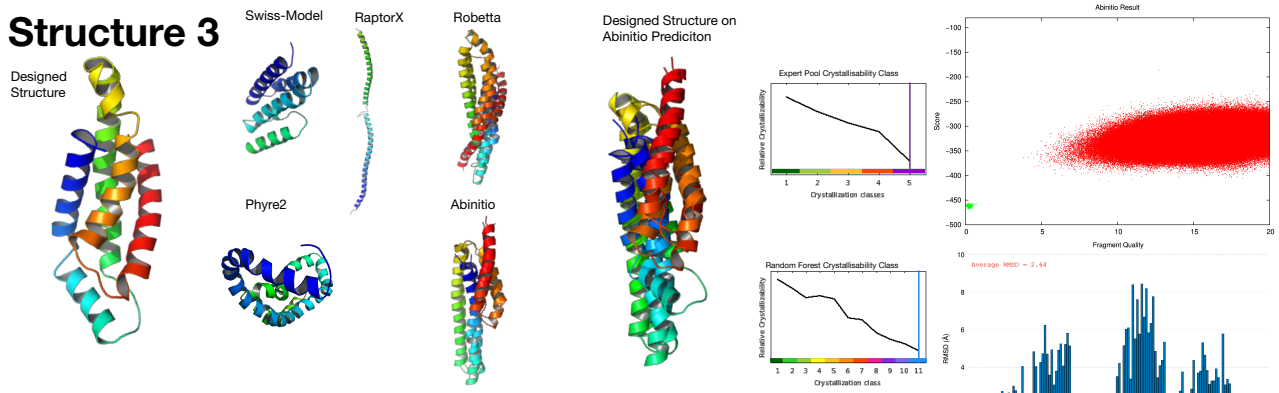
Structure 2



PSIPRED secondary structure prediction

```
Conf: 947899999999999999999999999999999999999875109999999999999999999999999999968985899999999999999943039864999999999999999999999999999999999873149
Pred: CCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCCHHHHHHHHHHHHHHHHHHHHHCCCCCCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
  AA: DGEIEKVAQEAAHEAAKAAEKAVKEAAEKAKEKAQKKMEQKKKTTEEVKKKKKEAGAAKQMQQRIAKKAKEAAEKLGGYDBAAAAVEKIVRIIEKVAEEKTGADDEEAAEELAKEAAAKKVKKIAEEHIEKKVKKKKEEKKKKKD
```

Structure 3



PSIPRED secondary structure prediction

```
Conf: 932788999999999999999999999999999983143458735899999999999999999999999999999999999999985743899999999999999999999664100129999999999999999999999709
Pred: CCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCCCCCCCCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCCHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
  AA: HGGEFAKKAIEEVAEKVKKKVVKEKVKEEAAKKIQKHGESTTGGSGAAKEAIEAAAAEAAIEAAAKKVQKWAKEAKKKEKEKAKKEVGTDKAVQEVAEKAVVQAAVEEVGEAKNAKTDIVAKVAEELAKEAAKKVKEQVKKQG
```

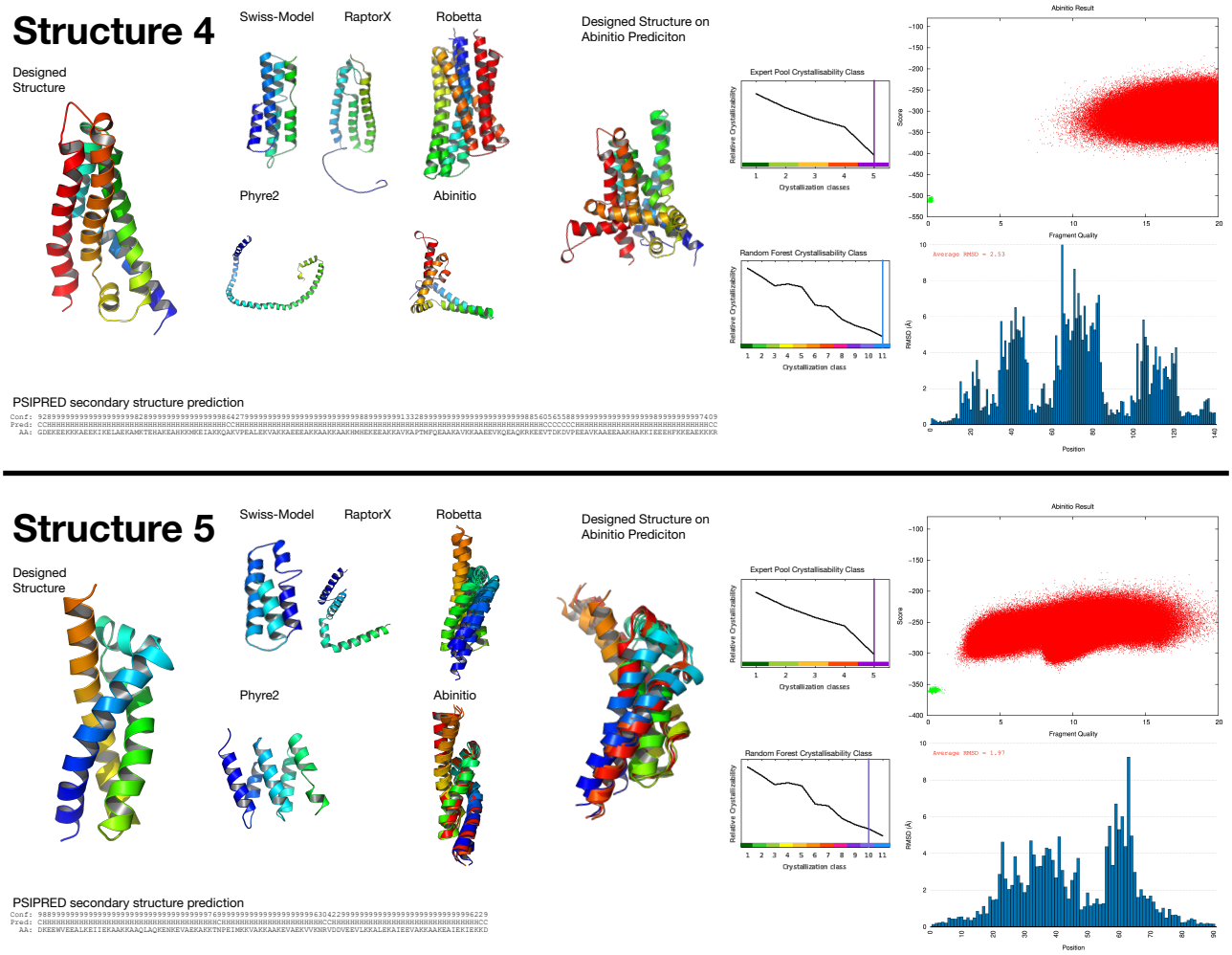


Figure 6: This figure summarises the results of each structure's predictions[36]. It can be seen that Swiss-Model, Phyre2, and RaptorX all failed to model the designed structure. While PSIPRED predicts the secondary structures at the correct places.

The local folding simulation using *Abinitio* showed almost funnel shaped plots, even though each structure was simulated more than 1 million times (red dots), *Abinitio* still failed to find energy states that are at the level of the designed structures (green dots). This can be logically determined because of the fragment quality (bars) that showed the best lowest RMSD of some positions (mainly loops) to still be greater than 4Å bring up the average best possible RMSD of the simulation to above 2Å. This was not a surprise since the fragments are derived from natural crystal structures and our topology was unnatural. Nonetheless, the local *Abinitio* and Robetta server simulations (both using the same protocol but with different optimisations) predicted structures that are very close to the designed structure, indicating the possibility of easily improving this system in the future. Structures 1 and 5 seem to have "OK" *Abinitio* simulations where the simulated structures are close to the designed structure from the neural network but the loops are folded incorrectly resulting in high RMSD. Structure 3 was moderately predicted, while structures 2 and 4 were very bad.

All structures seem to have intrinsic disordered states making predicting their crystallisability difficult, this was a result of the RosettaDesign algorithm rather than the topology of the neural network generated structure.

It is clear from these folding results in (figure 5) that more is needed to better sequence design the structures according to the backbone topology. The main issue we faced was that all designed structures were predicted to be intrinsically disordered proteins (IDP), which makes them difficult to model as well as difficult to crystallise (as determined from XtalPred-RF). Furthermore, the *Abinitio* folding simulation had issues with the fragments, as can be seen from the respective plots, some of the best fragments had high RMSD values (mainly at the loop positions) brining up the final possible simulated RMSD for the structure. On the other hand, all algorithms were good at predicting most helical positions in all the structures, highlighting the issue to be restricted to the loops positions.

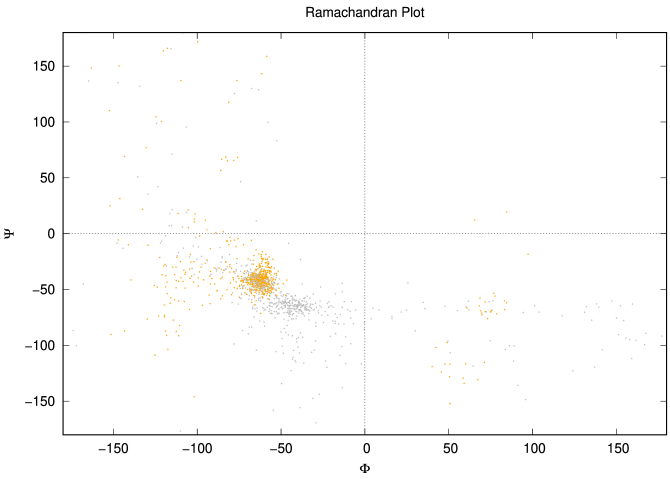


Figure 7: The network's output Φ and Ψ angles before and after the relax step, clearly showing a shift from a wide distribution of angles in the unrelaxed structures fresh out of the neural network (grey) to a concentration of ideal helical angles after the relaxation (orange). This is a culmination of all five structures in figure 6.

The logical next step is to synthesise and crystallise the proteins to confirm their structures.

Conclusion:

In this paper, we outlined how we developed a neural network architecture that can definitively design a helical and compact protein backbone, and how we added our network to the flexible backbone protocols to achieve *de novo* protein design. Then we attempted to computationally simulate and evaluate the realistic final fold of our designed structures which was not as successful and clear as we hoped, thus we could not determine whether the fault is with the design protocol or with the folding simulation. Thus, flxbb-design protocol should be optimised to result is less IDPs. We still lack a definitive way to evaluate our designed structures computationally, thus we should perform X-Ray/NMR of multiple proteins in the future. Nonetheless, the backbone topology from the neural network was good, we identified that using only the Φ and Ψ angles of a protein structure were adequate features to design a protein topology. Though our network had a low success rate (~0.40%), generating multiple structures and auto filtering the bad ones proved an adequate setup to achieve our goal of *de novo* helical protein design within a reasonable time (1-6 hours for the backbone topology and ~24 hours for the flexible-backbone design). Still the low success rate is not satisfactory, and we are currently working on an improved model the uses further dataset dimensions to increase the success rate. The neural network, along with the datasets and the trained weights, are available at this GitHub repository which includes an extensive README file and a video that explains how to use the scripts. As a next step, we will try to improve the neural network's output by implementing an additional feature (Cα distances - sometimes referred to as constraints) to generate better loops using the same network architecture or a different one, as well as crystallise the output structures to get a definitive understanding of the neural network capabilities.

Acknowledgement:
The corresponding author would like to thank the High Performance Computing Centre at King Abdulaziz University for making available the Aziz high performance computer where the corresponding author was able to augment the dataset and perform the *Abinitio* folding simulations.

References:

1. Huang, P. S., Boyken, S. E. & Baker, D.. 2016. The coming of age of de novo protein design. Nature 537: 320-327. doi: 10.1038/nature19946. PMID:27629638
2. Dougherty, M. J. & Arnold, F. H.. 2009. Directed evolution: new parts and optimized function. Curr Opin Biotechnol 20: 486-91. doi: 10.1016/j.copbio.2009.08.005. PMID:19720520
3. Alford, R. F., Leaver-Fay, A., Jeliazkov, J. R., O'Meara, M. J., DiMaio, F. P., Park, H., Shapovalov, M. V., Renfrew, P. D., Mulligan, V. K., Kappel, K., Labonte, J. W., Pacella, M. S., Bonneau, R., Bradley, P., Dunbrack, R. L., Jr., Das, R., Baker, D., Kuhlman, B., Kortemme, T. & Gray, J. J.. 2017. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. J Chem Theory Comput 13: 3031-3048. doi: 10.1021/acs.jctc.7b00125. PMID:28430426
4. Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L. & Baker, D.. 2003. Design of a novel globular protein fold with atomic-level accuracy. Science 302: 1364-1368. doi: 10.1126/science. 1089427. PMID:14631033
5. Huang, P. S., Ban, Y. E., Richter, F., Andre, I., Vernon, R., Schief, W. R. & Baker, D.. 2011. RosettaRemodel: a generalized framework for flexible backbone protein design. PLoS One 6: e24109. doi: 10.1371/journal.pone.0024109. PMID:21909381
6. Koga, N., Tatsumi-Koga, R., Liu, G., Xiao, R., Acton, T. B., Montelione, G. T. & Baker, D.. 2012. Principles for designing ideal protein structures. Nature 491: 222-227. doi: 10.1038/nature11600. PMID:23135467
7. Regan, L. & DeGrado, W. F.. 1988. Characterization of a helical protein designed from first principles. Science 241: 976-978. PMID:3043666
8. Harbury, P. B., Plecs, J. J., Tidor, B., Alber, T. & Kim, P. S.. 1998. High-resolution protein design with backbone freedom. Science 282: 1462-1467. PMID:9822371
9. Grigoryan, G. & Degrado, W. F.. 2011. Probing designability via a generalized model of helical bundle geometry. J Mol Biol 405: 1079-100. doi: 10.1016/j.jmb.2010.08.058. PMID:20932976
10. Huang, P. S., Oberdorfer, G., Xu, C., Pei, X. Y., Nannenga, B. L., Rogers, J. M., DiMaio, F., Gonen, T., Luisi, B. & Baker, D.. 2014. High thermodynamic stability of parametrically designed helical bundles. Science 346: 481-485. doi: 10.1126/science.1257481. PMID:25342806
11. Joh, N. H., Wang, T., Bhate, M. P., Acharya, R., Wu, Y., Grabe, M., Hong, M., Grigoryan, G. & DeGrado, W. F.. 2014. De novo design of a transmembrane Zn(2)(+)-transporting four-helix bundle. Science 346: 1520-1524. doi: 10.1126/science.1261172. PMID:25525248
12. Thomson, A. R., Wood, C. W., Burton, A. J., Bartlett, G. J., Sessions, R. B., Brady, R. L. & Woolfson, D. N.. 2014. Computational design of water-soluble alpha-helical barrels. Science 346: 485-488. doi: 10.1126/science.1257452. PMID:25342807
13. Wang, J., Cao, H., Zhang, J. Z. H. & Qi, Y.. 2018. Computational Protein Design with Deep Learning Neural Networks. Sci Rep 8: 6349. doi: 10.1038/s41598-018-24760-x. PMID:29679026
14. Tyka, M. D., Keedy, D. A., Andre, I., Dimaio, F., Song, Y., Richardson, D. C., Richardson, J. S. & Baker, D.. 2011. Alternate states of proteins revealed by detailed energy landscape mapping. J Mol Biol 405: 607-18. doi: 10.1016/j.jmb.2010.11.008. PMID: 21073878
15. Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L. & Baker, D.. 2003. Design of a novel globular protein fold with atomic-level accuracy. Science 302: 1364-1368. doi: 10.1126/science. 1089427. PMID: 14631033
16. Chaudhury, S., Lyskov, S. & Gray, J. J.. 2010. PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. Bioinformatics 26: 689-91. doi: 10.1093/bioinformatics/ btq007. PMID: 20061306
17. Koga, N., Tatsumi-Koga, R., Liu, G., Xiao, R., Acton, T. B., Montelione, G. T. & Baker, D.. 2012. Principles for designing ideal protein structures. Nature 491: 222-227. doi: 10.1038/nature11600. PMID: 23135467
18. Rohl, C. A., Strauss, C. E., Misura, K. M. & Baker, D.. 2004. Protein structure prediction using Rosetta. Methods Enzymol 383: 66-93. doi: 10.1016/s0076-6879(04)83004-0. PMID: 15063647
19. Gront, D., Kulp, D. W., Vernon, R. M., Strauss, C. E. & Baker, D.. 2011. Generalized fragment picking in Rosetta: design, protocols and applications. PLoS One 6: e23294. doi: 10.1371/journal.pone.0023294. PMID: 21887241

20. Song, Y., DiMaio, F., Wang, R. Y., Kim, D., Miles, C., Brunette, T., Thompson, J. & Baker, D.. 2013. High-resolution comparative modeling with RosettaCM. Structure 21: 1735-42. doi: 10.1016/j.str.2013.08.005. PMID: 24035711
21. Raman, S., Vernon, R., Thompson, J., Tyka, M., Sadreyev, R., Pei, J., Kim, D., Kellogg, E., DiMaio, F., Lange, O., Kinch, L., Sheffler, W., Kim, B. H., Das, R., Grishin, N. V. & Baker, D.. 2009. Structure prediction for CASP8 with all-atom refinement using Rosetta. Proteins 77 Suppl 9: 89-99. doi: 10.1002/prot.22540. PMID: 19701941
22. Bertoni, M., Kiefer, F., Biasini, M., Bordoli, L. & Schwede, T.. 2017. Modeling protein quaternary structure of homo- and hetero-oligomers beyond binary interactions by homology. Sci Rep 7: 10480. doi: 10.1038/s41598-017-09654-8. PMID: 28874689
23. Benkert, P., Biasini, M. & Schwede, T.. 2011. Toward the estimation of the absolute quality of individual protein structure models. Bioinformatics 27: 343-50. doi: 10.1093/bioinformatics/btq662. PMID: 21134891
24. Guex, N., Peitsch, M. C. & Schwede, T.. 2009. Automated comparative protein structure modeling with SWISS-MODEL and Swiss-PdbViewer: a historical perspective. Electrophoresis 30 Suppl 1: S162-73. doi: 10.1002/elps.200900140. PMID: 19517507
25. Bienert, S., Waterhouse, A., de Beer, T. A., Tauriello, G., Studer, G., Bordoli, L. & Schwede, T.. 2017. The SWISS-MODEL Repository-new features and functionality. Nucleic Acids Res 45: D313-d319. doi: 10.1093/nar/gkw1132. PMID: 27899672
26. Waterhouse, A., Bertoni, M., Bienert, S., Studer, G., Tauriello, G., Gumienny, R., Heer, F. T., de Beer, T. A. P., Rempfer, C., Bordoli, L., Lepore, R. & Schwede, T.. 2018. SWISS-MODEL: homology modelling of protein structures and complexes. Nucleic Acids Res 46: W296-w303. doi: 10.1093/nar/gky427. PMID: 29788355
27. Kallberg, M., Wang, H., Wang, S., Peng, J., Wang, Z., Lu, H. & Xu, J.. 2012. Template-based protein structure modeling using the RaptorX web server. Nat Protoc 7: 1511-22. doi: 10.1038/nprot.2012.085. PMID: 22814390
28. Kelley, L. A., Mezulis, S., Yates, C. M., Wass, M. N. & Sternberg, M. J.. 2015. The Phyre2 web portal for protein modeling, prediction and analysis. Nat Protoc 10: 845-58. doi: 10.1038/nprot.2015.053. PMID: 25950237
29. Buchan, D. W., Minneci, F., Nugent, T. C., Bryson, K. & Jones, D. T.. 2013. Scalable web services for the PSIPRED Protein Analysis Workbench. Nucleic Acids Res 41: W349-57. doi: 10.1093/nar/gkt381. PMID: 23748958
30. Alzantot, Moustafa, Chakraborty, Supriyo & Srivastava, Mani. 2017. Sensegen: A deep learning architecture for synthetic sensor data generation. IEEE. https://arxiv.org/abs/1701.08886
31. Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. & de Hoon, M. J.. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 25: 1422-1423. doi: 10.1093/bioinformatics/btp163. PMID: 19304878
32. Joosten, R. P., te Beek, T. A., Krieger, E., Hekkelman, M. L., Hooft, R. W., Schneider, R., Sander, C. & Vriend, G.. 2011. A series of PDB related databases for everyday needs. Nucleic Acids Res 39: D411-9. doi: 10.1093/nar/gkq1105. PMID: 21071423
33. Murphy, G. S., Mills, J. L., Miley, M. J., Machius, M., Szyperski, T. & Kuhlman, B.. 2012. Increasing sequence diversity with flexible backbone protein design: the complete redesign of a protein hydrophobic core. Structure 20: 1086-96. doi: 10.1016/j.str.2012.03.026. PMID: 22632833
34. Baakman, Coos, Krieger, E., Black, Jon, Touw, Wouter G., Vriend, Gert, Joosten, Robbie P. & te Beek, Tim A. H.. 2014. A series of PDB-related databanks for everyday needs. Nucleic Acids Research 43: D364-D368. doi: 10.1093/nar/gku1028. https://dx.doi.org/10.1093/nar/gku1028. PMID: 25352545
35. Slabinski, L., Jaroszewski, L., Rychlewski, L., Wilson, I. A., Lesley, S. A. & Godzik, A.. 2007. XtalPred: a web server for prediction of protein crystallizability. Bioinformatics 23: 3403-3405. doi: 10.1093/bioinformatics/btm477. PMID: 17921170
36. Schrodinger, LLC. 2015. The PyMOL molecular graphics system, version 1.8. Schrodinger LLC, New York, NY
37. Radford, Alec, Metz, Luke & Chintala, Soumith. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434