Update Ex4 network with a new capability - accuracy

Adi Bonofiel, Sagi Shaked, Yaniv Melamed, Guy Truzman

Introduction

Nanobodies are monoclonal, small and highly-stable antibodies that have been engineered to be smaller than conventional antibodies [1]. They are typically about one-tenth the size of a conventional antibody, and their small size confers several advantages. Nanobodies can more easily penetrate tissues, and they can bind to targets that are otherwise inaccessible to conventional antibodies. Nanobodies also have a longer half-life in the body and are less likely to provoke an immune response. Recently, researchers found out nanobodies can efficiently block microbes, such as SARS-Cov-2 [2].

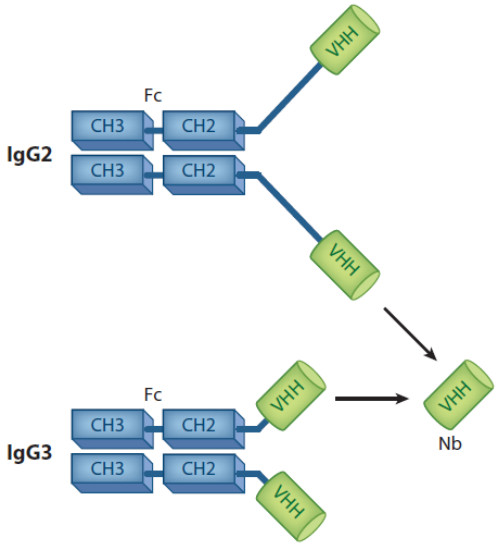

Fig1. Heavy-Chain Antibodies And Nanobodies (www.annualreviews.org).

Nanobodies have three complementarity-determining regions (CDR1, CDR2, CDR3). These regions form three variable loops and are the most important part of the antigen-nanobody interaction (and are also the most challenging parts to model - especially CDR3).

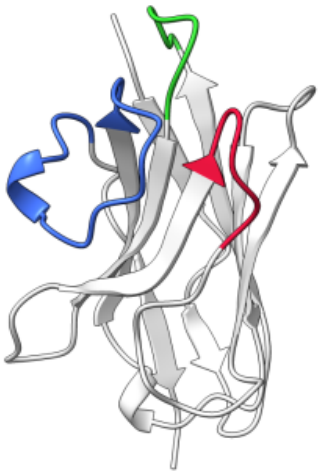

Fig2. Nanobody structure - CDR1 in green, CDR2 in red & CDR3 in blue.

The variability of these regions is responsible for the specificity of the nanobody. Therefore, in contrast to the other regions of the nanobodies, it is harder to predict the 3D structure of these regions.

In the past few years, deep learning has taken a significant part in answering fundamentally important questions in structural biology, such as protein folding. Today, the state-of-the-art AI neural network is AlphaFold2, launched by DeepMind. Nevertheless, AF2's predictions for nanobodies are not yet precise.

In Ex4, we trained a neural network with specific architecture only on nanobodies, trying to take advantage of the relatively similar regions of the nanobodies. In this project, we decided to take this network one step further and try to predict the accuracy for each position.

Accuracy score:

Scored poses with an RMSD of less than or equal to 1.5Å are considered to be successful [3]. Therefore, we decided to create a new relative score for accuracy, based on this score. Given R, a protein structure, and P, a prediction for the structure of this protein, the accuracy for the i-th residue in the prediction will be

$$Score(P_i, R_i) = max(\frac{3-RMSD(P_i,R_i)}{3}, 0) * 100$$

Where the RMSD is calculated as

$$RMSD(P_i, R_i) = \sqrt{(x_{P_i} - x_{R_i})^2 + (y_{P_i} - y_{R_i})^2 + (z_{P_i} - z_{R_i})^2}$$

In this way, predictions with RMSD > 1.5Å will get a score > 50. This score was tested against our prediction from Ex4 and it behaves according to the behavior we expect it to (Fig3.). That is, for the constant regions of the protein we get high accuracy, as opposed to the variable regions.

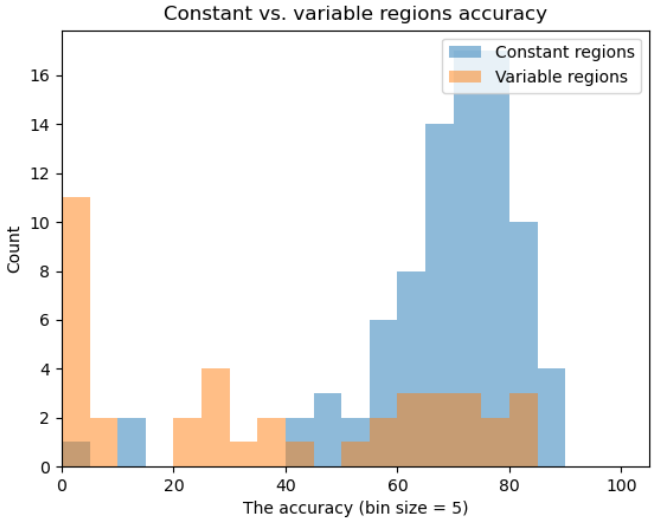

Fig3. Histogram of the constant regions' against the variable regions' accuracy.

Naïve accuracy calculation (baseline):

First, we decided to take a naive approach for accuracy calculation. We then generated 10 models, more or less accurate, with some architecture differences for each model (n=10 is the maximum number of models our program supports for now, based on the model in Table1). The amount of models our program supports can be changed for more/less models, and the architecture of each model can be changed as well, by providing the needed parameters to the program. For each protein, We used the models to produce 10 predictions (one prediction from each model) and translated these predictions into a distance matrix, based on RMSD. The prediction we chose for this from this naive model is the one with the minimal sum-of-RMSDs, as it behaves as the median between all 10 predictions. Then, for each residue, we calculated the mean above score between the chosen prediction's residue and all other 9 predictions as a predictor for the structure's accuracy.

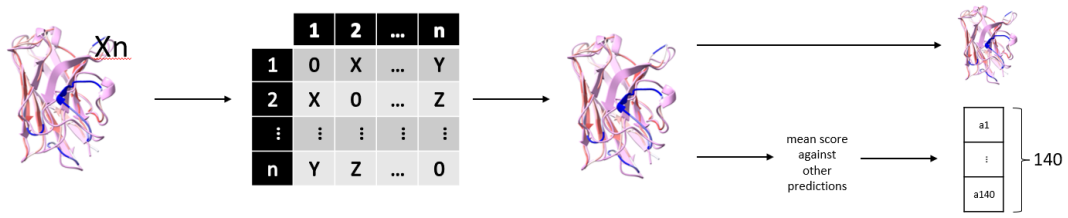

Fig4. The naive approach's workflow.

1. Predict structure with 10 different models.
2. Set the prediction with the lower RMSD vs. all the others as reference.
3. Output average RMSD vector and PDB file of the chosen reference.

Advanced accuracy calculation:

We modified the output and the training loop of pre-tested neural network architecture (i.e. EX4 network) which is known to predict structures properly, to generate accuracy prediction alongside the structure prediction.

To do so, the network output (i.e. last layer) contains two tensors - a tensor of structure prediction with shape of (BATCH_SIZE, 140, 15) and a tensor of accuracy of the predicted structure with shape of (BATCH_SIZE, 140) - i.e. accuracy score per each residue [4].

To obtain this modified functionality, the training loop utilizes two loss functions. The first one is simple RMSD loss between the real and the predicted structure. The output from this loss is used as an input to the second loss which measures the distance between the accuracy prediction of the network and the actual RMSD between the predicted and the real structure.

In the first epochs (n=10) we gave zero weight to the second loss, enabling the network to converge on accurate structure prediction. After this, we increase the weight of this loss (w=0.03) for the network to learn to use the second output as accuracy prediction. It was important to keep this weight small enough to prevent the network from generating inaccurate structure predictions and just predict small accuracy to those predictions.

Results & discussion

Our modified network structure prediction preserves the performance of the ex4 network (fig 1.a), which is not trivial since the additional loss tends to pull the network away from only structure prediction. As expected, the same applies for the baseline model (fig 1.b). A slight difference in the structure prediction precision observed between the baseline and modified network (fig 1.c), which converge with the training process of those two models, when the former implies to predict two different tasks while the latter focuses on minimizing one loss only.

Fig1. Models comparison

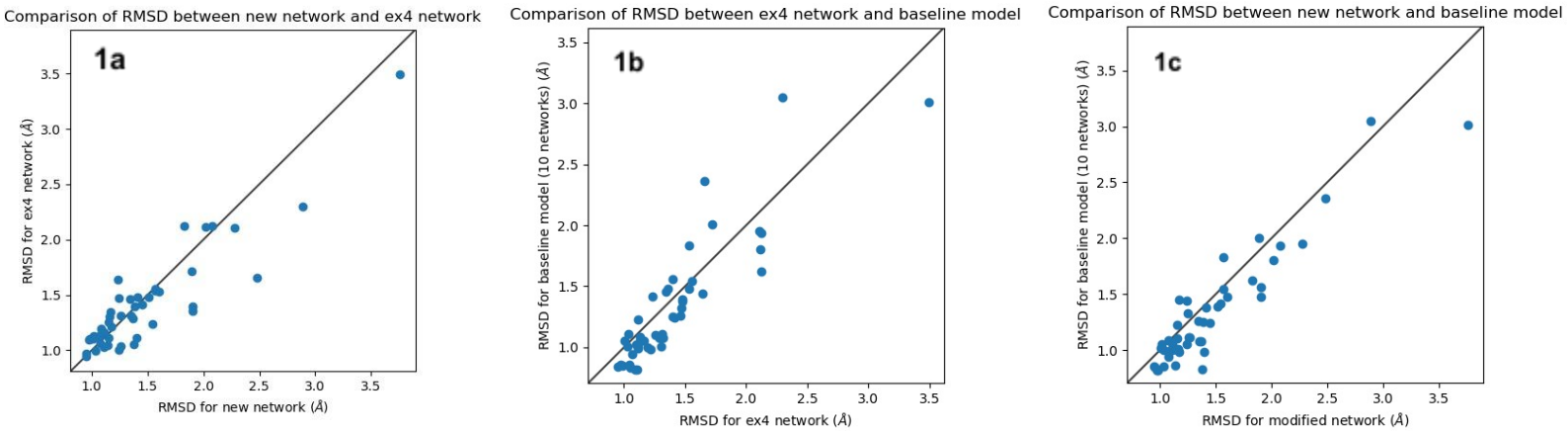

Fig 1. a. The RMSD score of each position in the advanced approach vs. ex4 network. **b.** The RMSD score of each position in the baseline approach vs. ex4 network. **c.** The RMSD score of each position in the baseline approach vs. advanced (on the test set).

In addition, as we can see in Fig 2. Below, our predicted accuracy is at some level close to the actual accuracy of the structure. We observe that some regions along the structure are predicted more accurately by one model and other regions are predicted more accurately by the other. In general, we can see that our modified network predicts lower accuracy than the actual accuracy of its predictions (figure 2a). However, the modified network predicts quite accurately its accuracy around the positions 97 to 105 in the structure (around the CDR3 region). In comparison to the modified network, the baseline model in general overestimates its accuracy compared to the actual accuracy of the prediction (figure 2b). We assume that the predicted accuracy by this model has higher estimated accuracy than the real accuracy, and the variance of the predicted accuracies are quite low because the 10 networks are quite similar in their architecture. We can see the networks are pretty different in their estimation of accuracy, as one (baseline model) overestimates the accuracy, and the other (modified model) underestimates

the accuracy (figure 2c). What we can also observe is that both networks are less confident when predicting the structure around the CDR3 region.

Fig2. Accuracy estimation

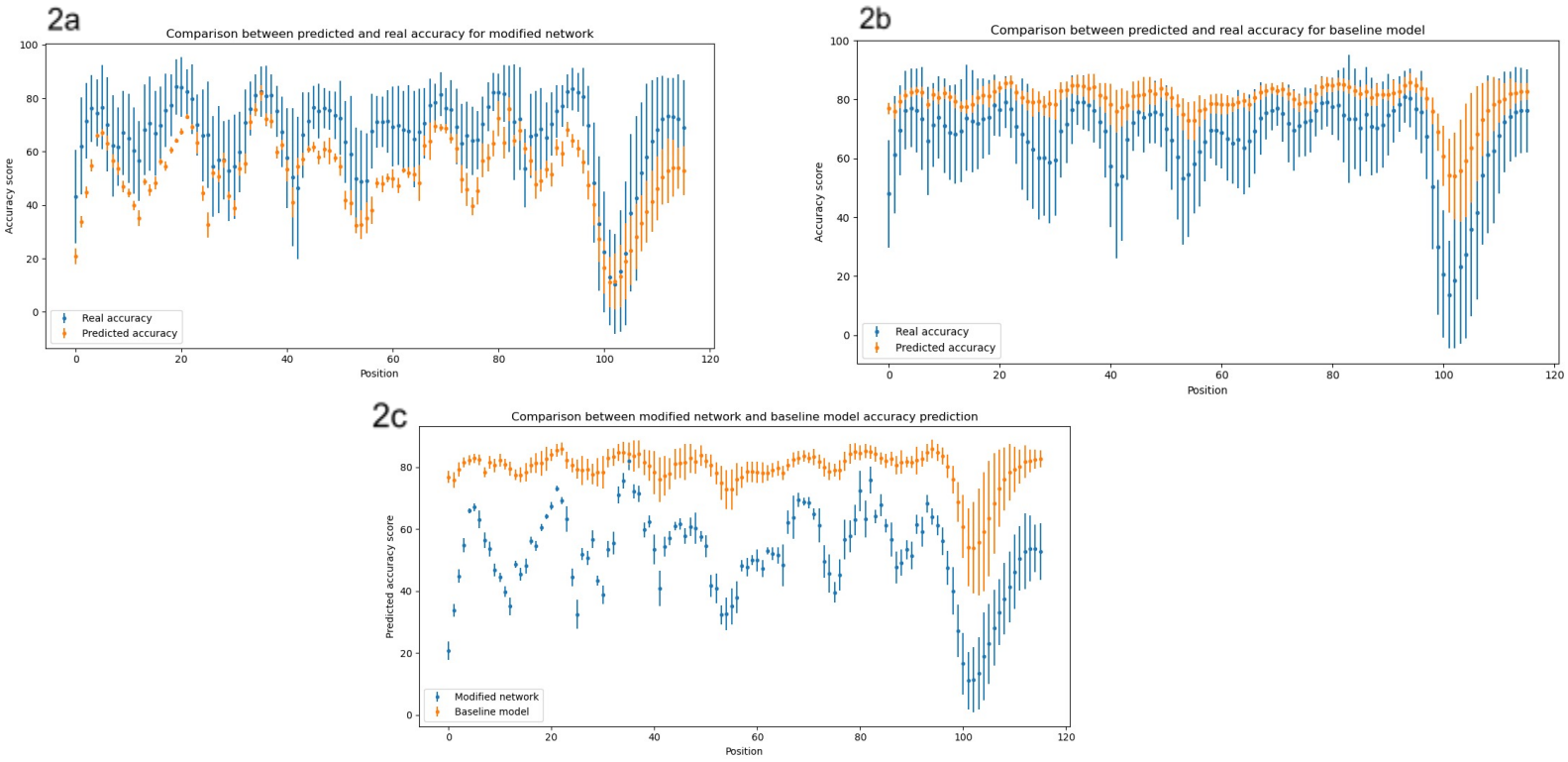

Fig. 2. a. Predicted accuracy with advanced model vs. real accuracy. **b.** Predicted accuracy with baseline model vs. real accuracy. **c.** Predicted accuracy with advanced model vs. predicted accuracy with baseline model (over the test set).

CDR3 chains tend to be the hardest part to model. Therefore, we thought to examine our model accuracy prediction on this part especially. Fig 3 shows comparison between CDR3 structure prediction and the true structure of two nanobodies from the test set - 3hc4 (fig 3.a), and 1gig (fig3.b). Indeed, the predicted accuracy of 3hc4 was high (rmsd=1.41 A) correlated with the actual high rmsd which was 0.98 A. In addition, the predicted accuracy of 1gig was low (rmsd=2.06 A) correlated (but slightly overestimated) with the actual low rmsd which was 5.64 A. This example emphasize that our model predicted accuracy represent an insightful information about the actual accuracy of the structure prediction

Fig3. CDR3 example

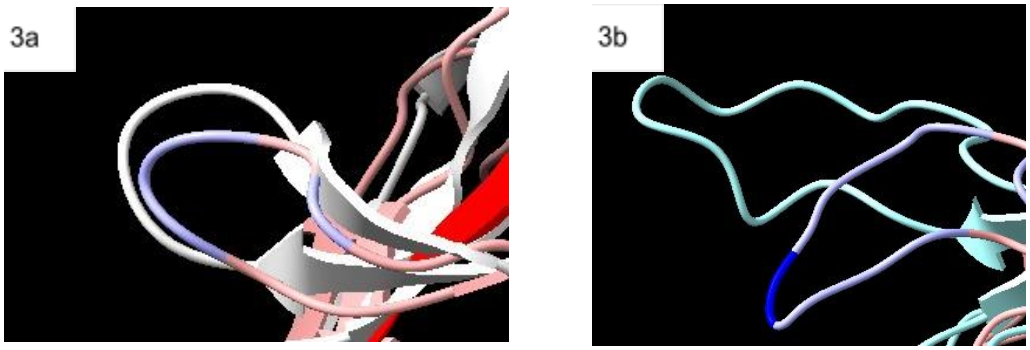

Fig. 3. a. Predicted and real CDR3 chain of 3hc4 file from the test set. **b.** Predicted and real CDR3 chain of 1gig file from the test set.

References

[1] M]muyldermans S. Nanobodies: natural single-domain antibodies. Annu Rev Biochem. 2013;82:775-97. doi: 10.1146/annurev-biochem-063011-092449. Epub 2013 Mar 13. PMID: 23495938.

[2] Nambulli S, Xiang Y, Tilston-Lunel NL, Rennick LJ, Sang Z, Klimstra WB, Reed DS, Crossland NA, Shi Y, Duprex WP. Inhalable Nanobody (PiN-21) prevents and treats SARS-CoV-2 infections in Syrian hamsters at ultra-low doses. Sci Adv. 2021 May 26;7(22):eabh0319. doi: 10.1126/sciadv.abh0319. PMID: 34039613; PMCID: PMC8153718.

[3] Hevener KE, Zhao W, Ball DM, Babaoglu K, Qi J, White SW, Lee RE. Validation of molecular docking programs for virtual screening against dihydropteroate synthase. J Chem Inf Model. 2009 Feb;49(2):444-60. doi: 10.1021/ci800293n. PMID: 19434845; PMCID: PMC2788795.

[4] Alex Kendall, Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? NIPS 2017, https://arxiv.org/abs/1703.04977

Supplementary links

Github, Drive

Supplementary material

	Amount of Resnet 1 blocks	Resnet 1 kernel size	Resnet 1 kernel number	Resnet 1 activation function	Amount of Resnet 2 blocks	Resnet 2 kernel size	Resnet 2 kernel number	Resnet 2 activation function	Dilations	Dropout value	Amount of epochs	Batch size	Dropout activation function	Learning rate
Model 1	3	15	64	ReLU	3	3	32	ReLU	[1,2,4,8]	0.15	60	32	ELU	0.01
Model 2	3	15	64	ReLU	5	5	32	ReLU	[1,2,4]	0.2	50	32	ELU	0.01
Model 3	3	15	64	ReLU	3	5	24	ReLU	[2,4,8,16]	0.25	60	64	ELU	0.01
Model 4	3	25	48	ReLU	2	5	32	ReLU	[1,2,4,8, 16]	0.15	60	32	ELU	0.01
Model 5	3	15	64	ReLU	5	5	32	GELU	[1,2,4,8]	0.2	60	32	ELU	0.01
Model 6	3	15	64	Softplus	3	3	32	Softplus	[1,2,4,8]	0.15	60	32	ELU	0.01
Model 7	3	15	64	ELU	5	3	64	ELU	[1,2,4,16]	0.2	60	32	ReLU	0.01
Model 8	3	15	64	ReLU	3	5	32	LeakyReLU	[1,2,4,8]	0.25	60	32	ELU	0.01
Model 9	4	15	64	SiLU	5	3	32	SiLU	[1,2,4]	0.2	50	64	ELU	0.01
Model 10	3	15	64	ReLU	2	5	64	ReLU	[1,2,4,8, 16]	0.25	120	32	ELU	0.001

Table 1: Parameters given to each model for building each network in 10nets model.