nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00803-z

Generating mutants of monotone affinity towards stronger protein complexes through adversarial learning

Received: 25 June 2023

Accepted: 6 February 2024

Published online: 28 February 2024

Check for updates

Tian Lan[1,2,3], Shuquan Su[1,2,8], Pengyao Ping[1,2,8], Gyorgy Hutvagner[4], Tao Liu[5,6], Yi Pan[7] & Jinyan Li[7] ✉

Despite breakthroughs achieved in protein sequence-to-structure and function-to-sequence predictions, the affinity-to-mutation prediction problem remains unsolved. Such a problem is of exponential complexity deemed to find a mutated protein or protein complex having a guaranteed binding-affinity change. Here we introduce an adversarial learning-based mutation method that creates optimal amino acid substitutions and changes the mutant's affinity change significantly in a preset direction. The key aspect in our method is the adversarial training process that dynamically labels the real side of the protein data and generates fake pseudo-data accordingly to construct a deep learning architecture for guiding the mutation. The method is sufficiently flexible to generate both single- and multipointed mutations at the adversarial learning step to mimic the natural circumstances of protein evolution. Compared with random mutants, our mutated sequences have in silico exhibited more than one order of change in magnitude of binding free energy change towards stronger complexes in the case study of Novavax–angiotensin-converting enzyme-related carboxypeptidase vaccine construct optimization. We also applied the method iteratively each time, using the output as the input sequence of the next iteration, to generate paths and a landscape of mutants with affinity-increasing monotonicity to understand SARS-CoV-2 Omicron's spike evolution. With these steps taken for effective generation of protein mutants of monotone affinity, our method will provide potential benefits to many other applications including protein bioengineering, drug design, antibody reformulation and therapeutic protein medication.

Point mutations, or their co-evolution, in protein amino acid sequences usually result in a protein folding into a different three-dimensional (3D) structure. Such structural changes have immediate impact on the protein's conformation and interaction stability with other proteins[1–5]. When binding affinity or binding strength, as measured by the binding free energy change of the complex following the mutation[6–10], becomes marked, the function of the mutant may be significantly enhanced by provoking changes in its binding affinity to receptors[11–13], or otherwise the mutant loses its original function.

This paper presents a machine learning method that makes an accurate prediction of a putative sequence from a given protein such that the mutated protein will have an in silico guaranteed

A full list of affiliations appears at the end of the paper. ✉e-mail: jinyan.li@siat.ac.cn

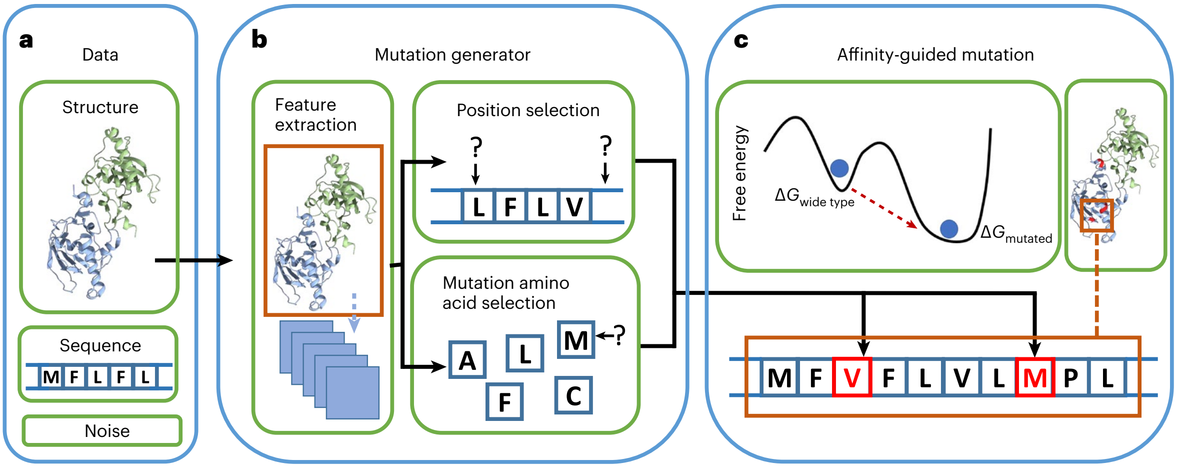

Fig. 1 | Overview of the DeepDirect mutation generator. a, Input data requirements. **b**, Mutation generation process. **c**, Binding affinity-guided mutation output.

binding affinity increase or decrease with the receptor. We call this an affinity-guided mutation, or an affinity-to-mutation prediction problem. The method can be applied iteratively to generate a path of mutated proteins having an increasing trend of monotone affinity in the iteration. Because interactions among proteins play critical roles in the basic functioning of cells and organisms, including immunity development, molecular transportation and metabolism[14,15], our method will be useful in many fields including protein engineering, protein structure determination, protein function prediction, drug design and protein evolution path construction, as has been seen from recent studies. For example, the Omicron BA.1 variant of SARS-CoV-2 can more easily escape from convalescent sera and monoclonal antibodies due to its lowered binding affinity compared with its earlier strains[16]. After acquiring stronger binding affinity to CD33 (ref. 17), M195, a monoclonal antibody, was found to have increased diagnostic and therapeutic capability for myeloid leukaemia; and the high-affinity programmed cell death protein 1 (PD-1) was found to be more effective than antiprogrammed cell death ligand 1 (anti-PD-L1) antibodies in the treatment of tumour in mouse models[18].

Exhaustive combination of random point mutations is a straightforward approach to solving this problem, but its exponential nature of computational complexity is too high to implement (an n-length amino acid sequence would have a total of 19^n potential combinations). We take deep learning as an efficient heuristic approach to narrow the search space through adversarial learning on the protein's specific structural data of receptor binding sites and learning on their atom properties to find potential mutation sites and generate the correct mutations at these sites. Tailored machine learning methods exist for sequence-to-affinity predictions, sequence-to-structure predictions and function-to-sequence predictions[19–36] (Supplementary Note 2). However, these methods are unable to answer our key questions: (1) which putative sequence will have an in silico guaranteed binding affinity increase and (2) what mutated sequence can reach maximum binding affinity with the receptor, namely the affinity-to-mutation prediction problem?

Generative adversarial network (GAN) is a type of generative deep learning framework proposed to solve generative modelling problems[37]. Its earlier variants, including conditional GAN[38] and deep convolutional GAN[39], were specially developed for a variety of generation tasks including image-to-image translation and text-to-image synthesis. However, these GAN algorithms suffer from problems such as unstable training process, vanishing gradients and mode collapse[40,41]. Wasserstein GAN (WGAN) improves performance by utilization of Wasserstein distance rather than Jensen–Shannon divergence implemented in the original GAN[42].

Based on the core concept of adversarial learning behind the generator–discriminator WGAN architectures, we present a new deep learning framework, DeepDirect, for generation of protein mutants under a preset affinity-increasing or -decreasing direction. The input of DeepDirect is the chain sequence *aaSeq* in a protein complex or that of the whole complex, and output $f(aaSeq)$ is a putative sequence mutated from *aaSeq* that has an increase or decrease in binding affinity with the receptor following the mutation.

DeepDirect is a framework that is able to generate mutations in protein amino acid sequences towards a change of direction in specified binding affinity. A novel step in our method is the adversarial training process that dynamically labels the real side of the data and generates fake pseudo-data accordingly to establish a computational model that guides the mutation generation. In addition to the classic generator and discriminator in a WGAN architecture, our DeepDirect architecture has a novel part, termed the binding-affinity change predictor. With coordination of the three parts, DeepDirect can select both mutation positions and amino acid substitutes according to the input protein's spatial information, leading to a direction-guided change in binding affinity. The model's flexibility in generating both single- and multipointed mutations partly mimics the natural circumstance of protein evolution.

It is of wider interest to find a putative sequence mutated from an existing protein such that it has maximum binding affinity with the receptor. We apply $f(aaSeq)$ iteratively using the output sequence each time as the input sequence of the next iteration, namely $f(\ldots, f(aaSeq))$, to reach stable status $f_n(aaSeq) = f_{(n+1)}(aaSeq)$, where $n \geq 1$ represents the number of iterations of $f(aaSeq)$. Then, the putative sequence $f_{(n+1)}(aaSeq)$ is a sequence mutated from aaSeq after n mutation steps that has a maximum affinity with the receptor. In fact, the affinities of $f_i(aaSeq)$, $i = 0, 1, \ldots, n$, shape a monotone increasing trend of change in binding free energy with the series of base mutations.

We draw a 3D landscape of binding affinities of those randomly mutated sequences and those sequences iteratively generated by our model $f_{(n+1)}(aaSeq)$ to illustrate how our algorithm effectively locates a peak point of binding affinities in the landscape and then jumps to a higher peak point. We use this affinity landscape to demonstrate how our deep learning algorithm overcomes the exponential complexity in the search space to find an optimal amino acid sequence that has maximum binding affinity. Specifically, these tests were conducted on the Novavax–angiotensin-converting enzyme-related carboxypeptidase (ACE2) complex to evaluate the effectiveness of DeepDirect's affinity-to-mutation prediction, and on the SARS-CoV-2 Omicron virus

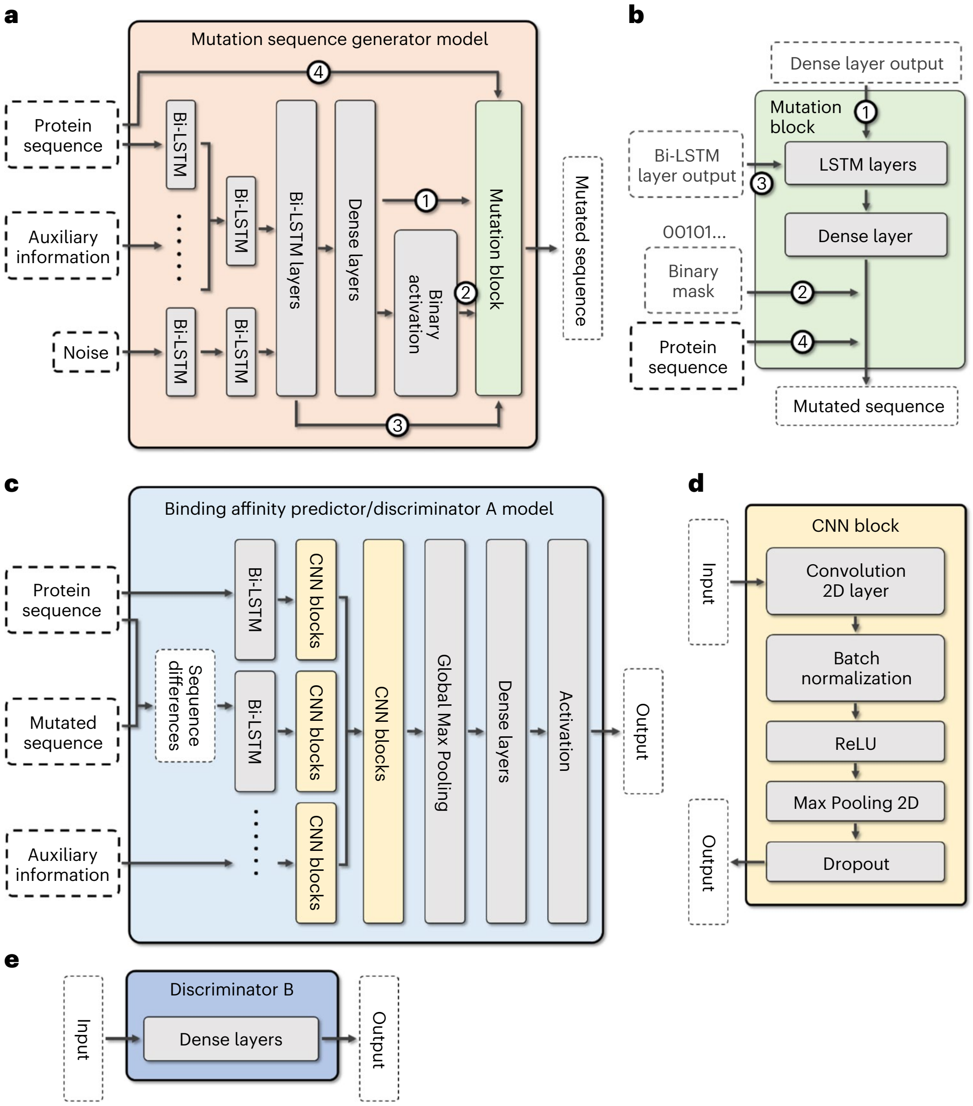

Fig. 2 | DeepDirect architecture. a, Mutation sequence generator. **b**, Mutation blocks inside the sequence generator. **c**, Binding-affinity predictor and discriminator A. **d**, CNN blocks inside the binding-affinity predictor and discriminator A. **e**, Discriminator B.

spike protein to predict a potential evolution path for the virus in terms of its interaction with the human ACE2 receptor.

Results

Architecture and adversarial training scheme of DeepDirect

DeepDirect has three main components: a protein sequence mutation generator, two discriminators and a protein complex binding-affinity change predictor.

The sequence mutation generator is the most important part of DeepDirect's architecture (Figs. 1 and 2a). Three types of data are required as input to the generator: the protein's amino acid sequence, the protein's structure/auxiliary data and protein-related noise (Fig. 1a). The mutation generator, together with the two discriminators and the binding affinity predictor, are organized in a new two-stage adversarial training procedure for the mutation generator to extract the required features. Benefiting from its architecture, the mutation generator is capable of determining mutation sites and the amino acid substitutions at these sites (Fig. 1b), towards a directed change in binding affinity (Fig. 1c).

The mutation block (Fig. 2b) is essential in DeepDirect to generate amino acid mutations with flexibility. The masking layer (Fig. 2a, flow 2) ensures the mutation sites to be selected with a flexible length based on features extracted from the input sequence, its auxiliary information and random noise. Combined with the extracted protein information (Fig. 2a, flows 1, 3 and 4), a mutated amino acid can be selected based on the input sequence from a 20-dimension space (corresponding to all potential amino acid substitutes) for each determined mutation position. As such, base mutations generated by DeepDirect are not limited to a single position, having a diversity of amino acid substitution patterns jointly affecting binding affinity.

DeepDirect has a two-stage training scheme for generation of expected mutations (Fig. 3). At stage A we train the mutation generator to produce appropriate mutations based on the properties of the protein sequence. Here, an appropriate mutation is one that, by avoiding the generation of an extra number of mutation positions, may lead to a totally different protein sequence. At stage B we train the mutation generator to generate mutations guided by a specific affinity-changing direction. We make use of two protein–protein interaction databases, AB-Bind[43] and SKEMPI v.2.0 (ref. 44), for the training. AB-Bind and SKEMPI v.2.0 contain 1,102 and 7,085 protein mutants, respectively, together with their experimentally determined binding free energy changes ($\Delta\Delta G$ (DDG), in kcal mol^{-1}). We note that all DDG data were split

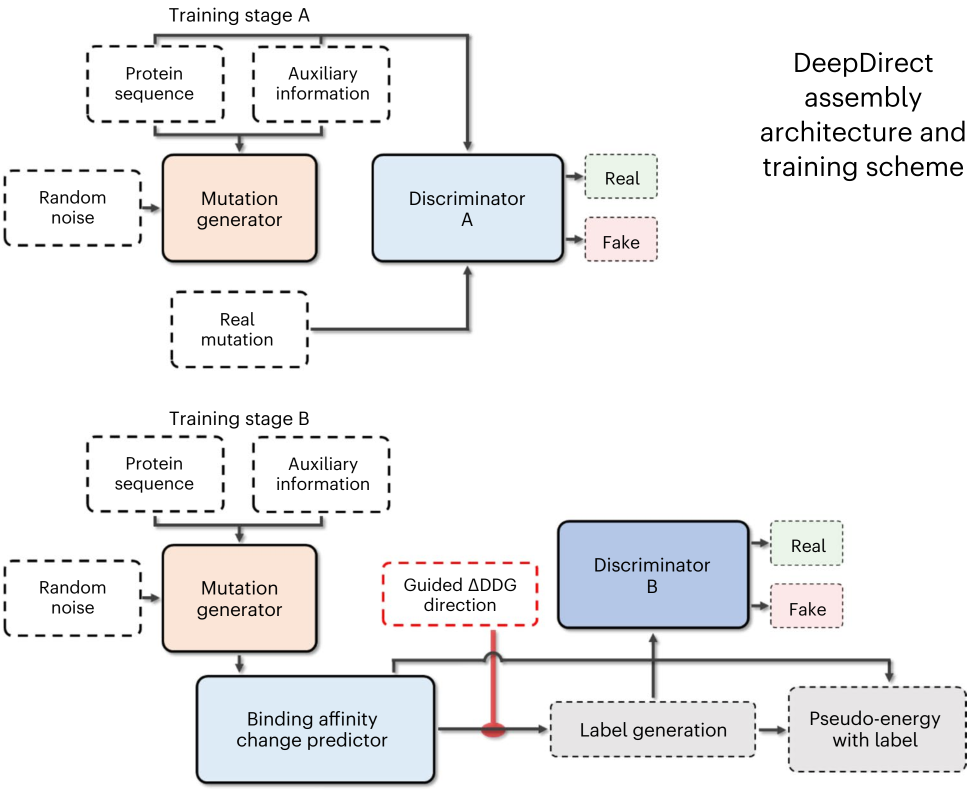

Fig. 3 | DeepDirect architecture and training scheme. Architecture and training scheme of training stage A (above) and B (below).

into 'train', 'validation' and 'test', in a ratio 0.7:0.15:0.15, for generalization validation of the binding-affinity change predictor. Further details regarding the training scheme are referred to in 'DeepDirect training framework and parameter settings'.

Deepdirect applied to a Novavax vaccine construct

One order of magnitude greater DDG change by DeepDirect. We evaluated the effectiveness of DeepDirect's affinity-guided mutation in comparison with random mutations. We randomly generated 1,500 mutated sequences of the Novavax vaccine for each of mutation rates 5, 10 and 20%; we also applied both models of DeepDirect—namely model$_{inc}$ (model trained to generate affinity-increasing mutations) and model$_{dec}$ (model trained to generate affinity-decreasing mutations)—to generate 500 mutated sequences using the Novavax–ACE2 complex sequence as input. More detailed parameter settings for the two models can be found at 'DeepDirect training framework and parameter settings'. The randomly mutated sequences have an average DDG of 0.24, 0.18 and 0.057 with a median 0.16, 0.09 and 0.006 under mutation rates 5, 10 and 20%, respectively (Supplementary Fig 1a). The 500 sequences generated by model$_{inc}$ have an average DDG of − 2.501 and median of −2.515, and the 500 generated by model$_{dec}$ have an average DDG of 5.352 and median of 5.312 (Fig. 4a). We additionally compared the the results from DeepDirect with those randomly mutated sequences grouped with a stronger/weaker binding affinity, as well as with chain-independent random mutations. The results demonstrate that DeepDirect's mutation mechanism is very effective in generating a new sequence that has one order of magnitude greater binding affinity increase or decrease than random mutants' affinity change (further comparison details can be found in Supplementary Note 3).

We note that only random mutations were compared, because DeepDirect is a method proposed to generate binding affinity-guided protein mutants whereas, biologically, every base in the input sequences allows a potential mutation. We found no models in the literature similar to ours in regard to performance benchmarking.

Vaccine optimization by model$_{inc}$. Furthermore, we reconstructed the Novavax vaccine trimer using model$_{inc}$ for in silico strengthening of the binding affinity of the vaccine construct with the human ACE2 receptor. Such an in silico reconstructed vaccine construct would render immune response more easily activated (in terms of binding to ACE2). In steps, we applied model$_{inc}$ to create mutated chain A of the Novavax–ACE2 complex with a batch size of 500 and then used the mutations on chain A of Novavax as template to reconstruct the other two chains and integrate the three chains as a trimer. Figure 4c shows that the reconstructed trimers have an average DDG of −2.44 kcal mol^{-1} with a median of −2.41 according to our prediction.

We also applied model$_{dec}$ and the random mutation approach under mutation rates of 5, 10 and 20% to reconstruct the vaccine construct with the same steps as above for comparison. Average changes in binding affinity for the reconstructed vaccine complexes by model$_{dec}$ and those by the random mutation approach under rates of 5, 10 and 20% were 6.14, 0.22, 0.11 and 0.11, with a median 6.00, 0.13, 0.04 and 0.03, respectively (Fig. 4c and Supplementary Fig 1b. Figure 4d illustrates one case of mutated positions among DeepDirect-generated mutants, where the overall stability of the complex has been enhanced by the mutation. Such mutants would be considered as potential candidates for strengthening the immunogenicity of the Novavax vaccine because of its capability of forming more stable conformation structures with the ACE2 receptor according to our prediction.

Because DeepDirect's generator requires receptor binding domain (RBD) information for the mutation task, our method also provides an inbuilt RBD prediction function for generating RBD index in situations where such information is lacking. We examined this prediction performance in the Novavax–ACE2 complex, finding that the predicted RBD area of the complex was located at the junction between the Novavax construct and ACE2 receptor, exactly as expected (Fig. 4e).

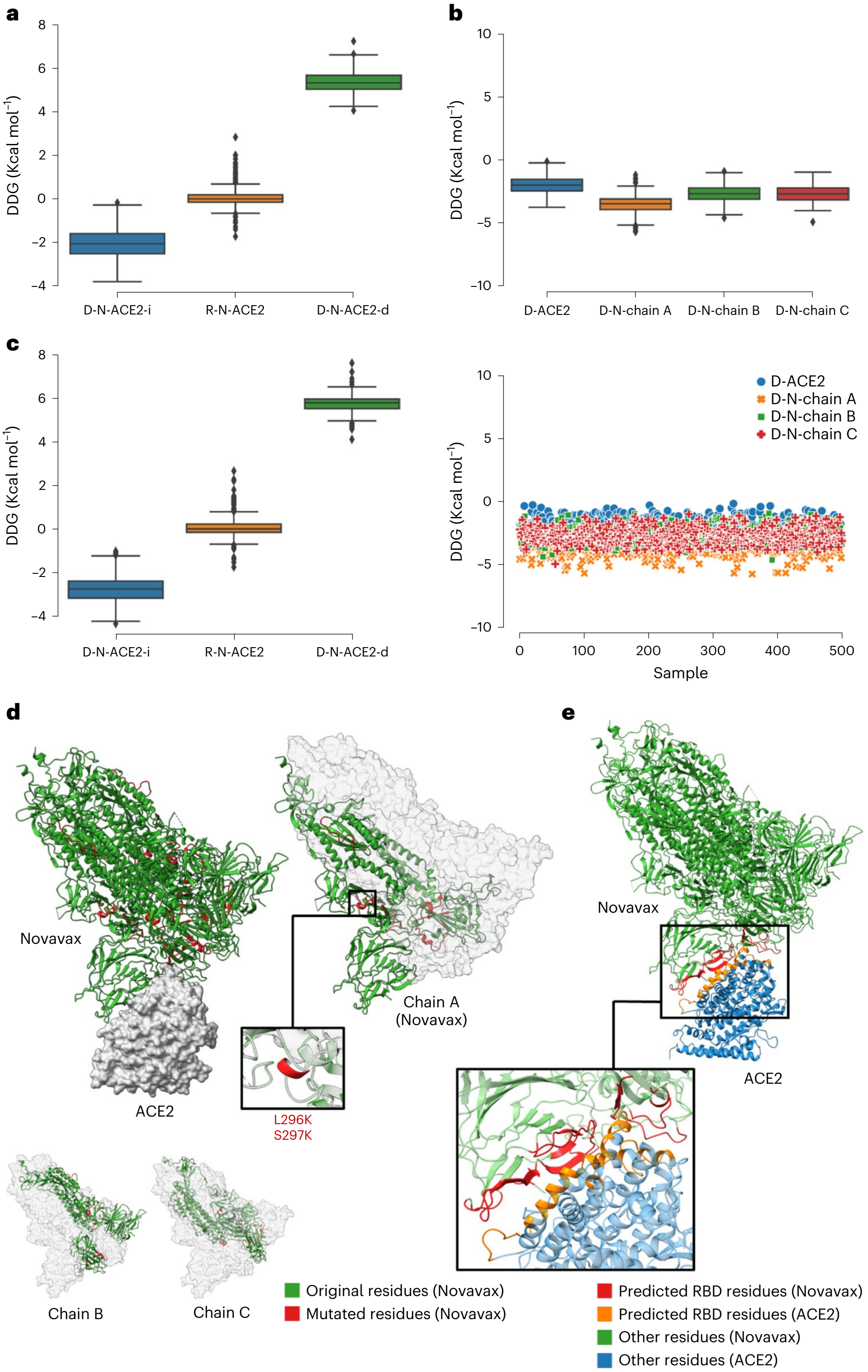

Fig. 4 | DDG comparison between randomly and DeepDirect-generated mutations for the Novavax–ACE2 complex. **a**, Boxplot of change in binding free energy between DeepDirect- and randomly generated mutations on the Novavax–ACE2 complex. $n = 500$ independent samples for each condition. **b**, Boxplot (top) and scatter plot (bottom) of change in binding free energy change of DeepDirect-generated mutations (binding affinity increase) on the Novavax–ACE2 complex (subset by chain). $n = 500$ independent samples for each condition. **c**, Boxplot of change in binding free energy between DeepDirect- and random mutation-reconstructed Novavax–ACE2 complex. $n = 500$ independent samples for each condition. **a**–**c**, Boxplots show median, first and third quartiles and minimum and maximum. Outliers are classified as being 1.5-fold outside the interquartile range. D-N-ACE2-i, R-N-ACE2, D-N-ACE2-d represent mutations on Novavax-ACE2 complex via Deepdirect model$_{inc}$, random process and Deepdirect model$_{dec}$. D-ACE2, D-N-chain A, D-N-chain B and D-N-chain C represent specific mutations on ACE2, and chain A, B and C of Novavax, via Deepdirect model$_{inc}$. **d**, DeepDirect mutated amino acid positions on Novavax chains A, B and C. **e**, DeepDirect-predicted RDB areas on the Novavax–ACE2 complex.

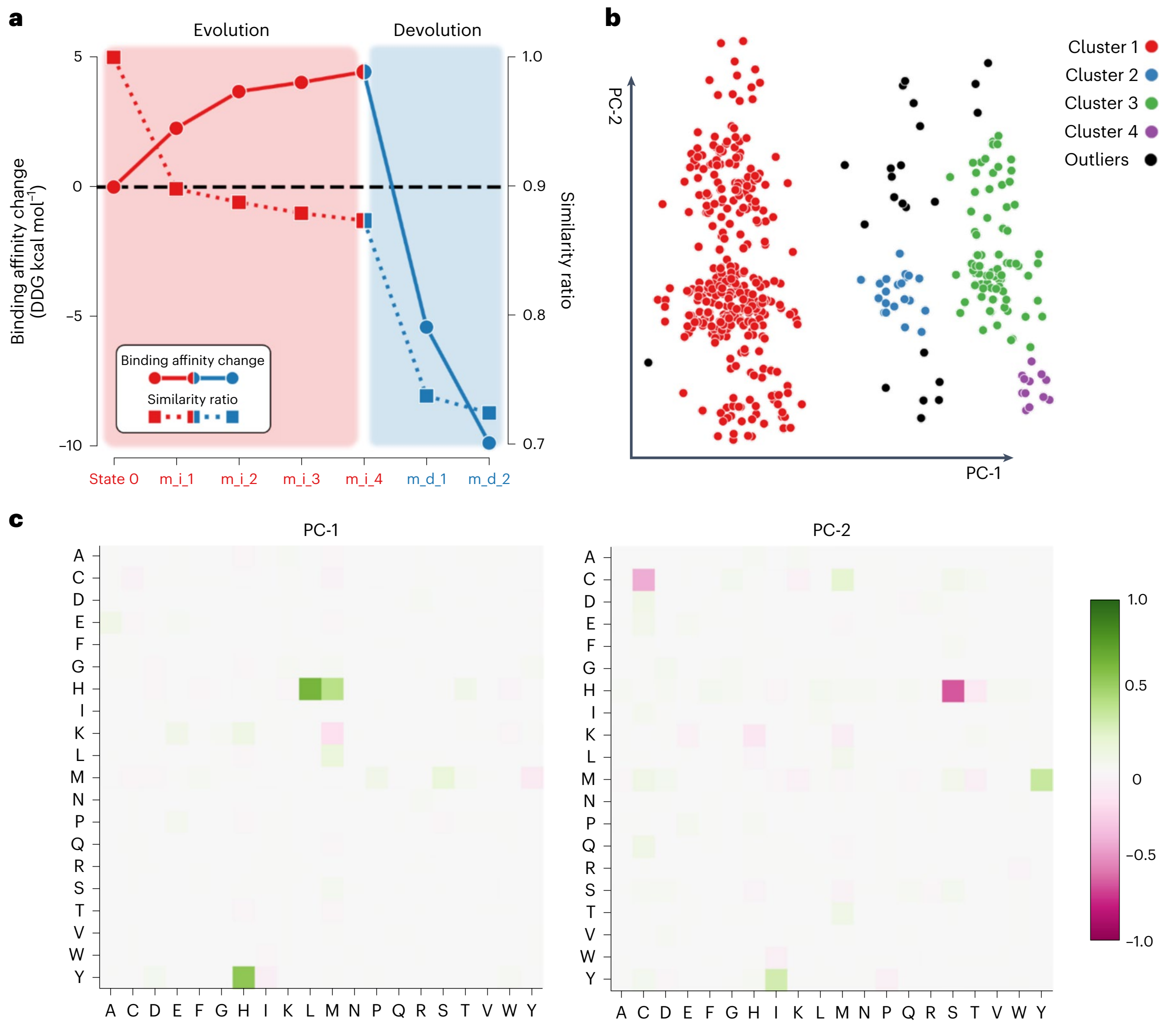

Fig. 5 | Putative evolution analysis of SARS-CoV-2 Omicron spike protein. **a**, Putative simulation of evolution and devolution paths. m_i/m_d represents mutating via model$_{inc}$/model$_{dec}$ and the following number indicates the mutation round. **b**, DBSCAN clusters with regard to the first two principal components (PC-1 and -2). **c**, Heatmap of the first and second principal components from SGT embeddings.

Omicron spike protein: evolution path and affinity landscape

We were interested in strain SARS-CoV-2 Omicron's spike protein and its potential evolution path. We applied model$_{inc}$ to the original sequence (denoted as s_0) of the spike protein to obtain putative sequence s_1, namely $s_1 = \mathrm{DM}_{inc}(s_0)$, where DM$_{inc}$ denotes the DeepDirect model$_{inc}$ mutation function. DeepDirect recommended 108 residue mutations (out of 1,061) for s_0. Following the mutation, the binding affinity of mutant s_1 was increased by 2.43 kcal mol^{-1} with the receptor (that is, complex free energy was reduced by 2.43 kcal mol^{-1}).

Affinity monotonicity. Iteratively we applied model$_{inc}$ each time, taking output sequence s_i as the input data to DM$_{inc}$(s) to obtain the next putative sequence, $s_{(i+1)}$. The iteration was stopped at s_4; The binding affinity of s_4 with the receptor showed little change compared with that of DM$_{inc}$(s_4), indicating that a potential maximum level of binding strength had been reached. With a total of 134 residue mutations (out of 1,061) from the original sequence s_0 in the four steps, putative sequence s_4 had gained 5.96 kcal mol^{-1} binding free energy with the ACE2 receptor (Fig. 5a).

On the other hand, we applied model$_{dec}$ with s_4 as input. As expected, the resulting sequence s_5 has a much weakened binding affinity with the receptor in comparison with that of s_4. We applied model$_{dec}$ repeatedly and obtained putative sequence s_7 whose binding strength was far less than the original level of that between s_0 and ACE2 (Fig. 5a). Interestingly, x_6 was not identical to x_0 in sequence, suggesting that there may be many unknown variants of Omicron's spike protein that have the same level of binding strength with ACE2. These results also suggest that SARS-CoV-2 Omicron variants may not have sufficient potential to mutate into a strong variant with significantly higher binding affinity to the ACE2 receptor.

There are two monotone trends of binding affinity as shown in Fig. 5a. One is an increasing trend when model$_{inc}$ was applied to s_0, the other a decreasing trend when model$_{dec}$ was applied to s_4. We term these, together with their associated mutants, either an in silico evolution or devolution path of the original SARS-CoV-2 Omicron spike protein sequence s_0. For the former path, mutant binding affinity increases sharply at the first mutation step and then slows down gradually in the remaining steps. However, as observed for the devolution path, binding affinity weakened much more rapidly than the increase in binding affinity during the evolution iterations.

Affinity landscape. To further understand unknown clusters of variants into which the SARS-CoV-2 Omicron virus might have evolved, we generated mutations 500 times separately by model$_{inc}$ on its spike protein sequences in a batch size of 500, to create 500 evolution paths. We then embedded the final 500 mutated sequences into vectors

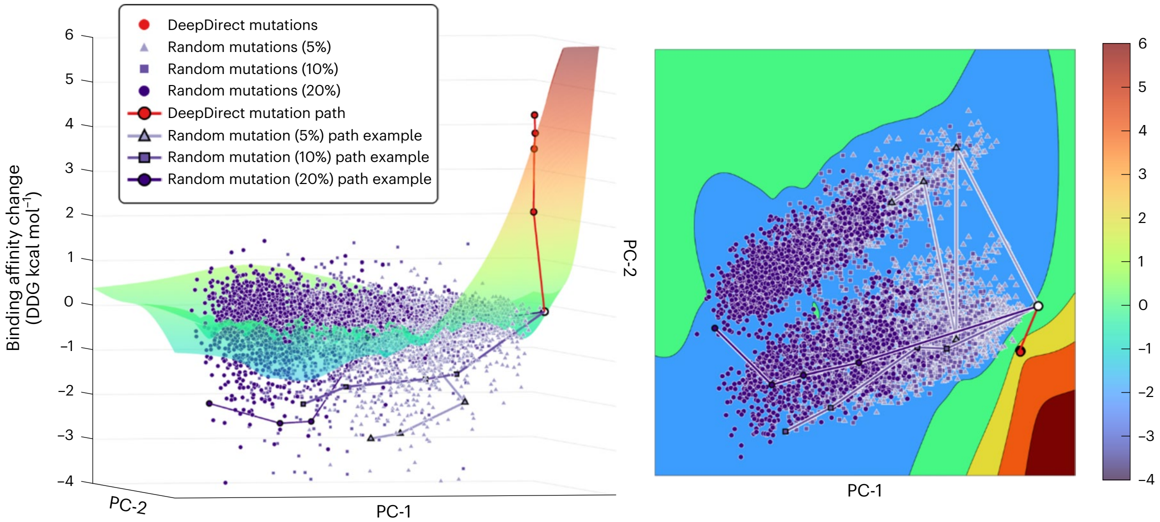

Fig. 6 | 3D landscape of binding affinities of SARS-CoV-2 Omicron spike mutants generated by DeepDirect and those generated by random mutations. Side (left) and top view (right) of the 3D landscape. Only some of the mutation paths are shown.

by a sequence graph transform (SGT) method[45] and further applied DBSCAN[46] on their first two principal components (PCs) for clustering of protein mutants. Four groups of sequences were formed in the hyperspace of the first two PCs, labelled 1–4, containing 368, 21, 76 and 12 mutants, respectively (Fig. 5b). DBSCAN also identified some outliers, showing that their sequences are less similar in terms of their extracted features compared with other sequences clustered into each group. Figure 5c shows the impact of the SGT (amino acid pair impact) in the first two PC spaces. Pair mutations H–Y, L–H and M–H (for PC-1) and H–S, M–Y and Y–I (for PC-2) have high absolute values, indicating their potentially key role in determininging the evolution paths of SARS-CoV-2 Omicron spike protein.

Figure 6 and Supplementary Fig 2b–d show a 3D landscape of binding affinities for randomly mutated sequences, as well as for those sequences generated by model$_{\text{inc}}$ from the Omicron spike protein sequence. The x–y plane of the 3D landscape is a hyperspace of the first two PCs of protein sequence embeddings obtained by SGT[45]. The z-axis of the landscape represents the binding affinities (DDGs) of protein mutants with the receptor. We compared the above DeepDirect-generated evolution path (four-step mutations towards stronger binding affinity) with three batches of 500 randomly generated mutation paths (each with the same four-step mutations) under mutation rates of 5, 10 and 20%. DeepDirect can generate mutations towards higher binding affinities whereas most randomly generated mutations go in different directions and eventually the sequences mutate toward a complex conformation of lower stability. This landscape also signifies that none of the randomly generated sequences has a binding affinity exceeding those of the putative sequences generated by our deep learning method. This verifies the effectiveness of the generator–discriminator adversarial learning concept as a heuristic idea aimed at narrowing the exponential search space to determine the maximum peak points of binding affinities from monotone increasing trends.

Discussion

We present DeepDirect, a deep learning framework for the generation of mutants from protein complexes with a specified direction of binding-affinity change so that mutants become either more or less stable. DeepDirect is an in silico approach used for the generation of affinity-guided mutations. As shown in the evaluation results, DeepDirect shows good performance in the detrmination of mutation sites that affect the binding free energy of the whole complex. As seen in Fig. 6a,b, DeepDirect is able to determine monotone paths in contrast to the random mutation approach. In addition, the model has the capability to deal with large-batch mutation generation tasks within a reasonable computing time.

The framework is implemented as a modified WGAN structure with three main components: a mutation generator, a discriminator and a predictor of binding-affinity change. We developed a new two-stage training scheme by first training the model to generate reasonable mutations from the reference, mimicking the natural mutation, and then learning to mutate along change in binding affinity direction. We designed such an objective-separated training scheme to help model enhanced extraction of required features, thereby improving training efficiency.

We demonstrated the effectiveness and application potential of DeepDirect by generating mutants for the Novavax vaccine construct. All 500 mutants generated by DeepDirect were found to have significantly stronger binding affinity compared with the random mutation process. In addition, by using DeepDirect to simulate the evolution paths for the SARS-CoV-2 Omicron virus spike protein, we found that the limited potential of the virus had evolved into a much stronger strain in terms of binding affinity to the ACE2 receptor. Four main groups into which the virus might have evolved were also predicted, as well as those amino acids that play key roles in that evolution. In addition to these case studies, DeepDirect has a wide range of application domains, providing researchers with efficient ways to better understand protein bioengineering and protein–protein interactions (see Supplementary Note 1 for further examples).

Note that we used DeepDirect to generate mutations based only on the initial protein conformation during its iterations for the SARS-CoV-2 Omicron spike protein. However, in reality those mutations in each iteration may also result in changes in the structure of the complex. The reason we did not consider those conformational changes is that there are currently few in silico protein structure prediction methodologies that have the capability to quantify those changes from mutation events (that is, the template-based searching strategy used by Alphafold prevents it from detecting minor structural changes from slight change in amino acids). We also note that all binding-affinity changes presented in the study were in silico predicted from DeepDirect's predictor of binding-affinity change, which is trained on protein mutation datasets containing wet-lab experimentally determined changed values of binding affinity. However, for more accurate detection of change in binding affinity, further wet-lab experiments may need to be carried out.

The current version of DeepDirect allows only substitution mutations in the input protein sequences but no insertion or deletion mutations, the main reason being that there are few data available relating to insertion or deletion mutations with change in labelled binding affinity for construction of the training model. Other challenges include: (1) how to define the length of continuous insertions or deletion in which an overmutation problem will occur and (2) how to handle overlap between insertion and deletion areas. There is no doubt that consideration of insertions and deletions will widen the mutation search space for DeepDirect in the search for better mutants. This will be a future development in upgrading DeepDirect. We also note that recently emerged foundation models for protein-related prediction tasks (for example, by ESM-2) have shown their superior performance. Integration of these models may of benefit in regard to DeepDirect's performance, which is another potential development area in our future work.

Methods

Data preprocessing

The $\Delta\Delta G$ data used in this study were derived directly from the AB-bind database. For the SKEMPI2 database we calculated $\Delta\Delta G$ values for each entry using

$$\Delta\Delta G = \Delta G_{\mathrm{mut}} - \Delta G_{\mathrm{wt}} \quad (1)$$

with

$$\Delta G_{\mathrm{mut}} = -RT\ln(K_{\mathrm{mut}}) \text{ and } \Delta G_{\mathrm{wt}} = -RT\ln(K_{\mathrm{wt}}) \quad (2)$$

where R is a universal gas constant equal to 8.314/4,184 (kcalK^{-1} mol^{-1}), T is set as 273.15 + 25(K) and K_{mut} and K_{wt} are, respectively, the binding affinity data after and before the mutation from each entry in the SKEMPI2 database. Mutated sequences were generated from the original sequences based on the mutations denoted in each database.

We used the alpha-carbon 3D coordinates of each amino acid as the coordinates of that amino acid from the protein's Protein Data Bank (PDB) file. The receptor–ligand index was constructed to distinguish receptor chains and ligand chains in the protein complex. We constructed the RBD index by locating the k (set as 50) nearest amino acids for each amino acid and counting the number among those k amino acids with a different receptor–ligand index. All counts were normalized, and those scores exceeding a cutoff (set at 0.1) were predicted as the amino acid at or around the RBD area. Amino acid sequences were one-hot encoded into a 20-element vector as input to a neural network.

Further details on DeepDirect architecture

A mutation generator, two discriminators and a binding affinity predictor are included in DeepDirect, as shown in Fig. 2.

The mutation generator is designed to generate binding affinity-guided mutations. Three types of data are required as input to the generator: the protein's amino acid sequence, its structure/auxiliary information and protein-related noise (Fig. 1a). Auxiliary information is defined as the index of the RBD of the protein complex, the amino acid corresponding to the ligand and the receptor in the complex, and the 3D coordinates information for each amino acid. Noise is generated using a Gaussian distribution. Each input data vector first undergoes bidirectional long short-term memory (bi-LSTM) layers separately for extraction of input information, and for the model to incorporate the input with different lengths. We set the latent dimension of all bi-LSTM layers at 128.

All bi-LSTM layer outputs are concatenated except for noise, and both the concatenated features and those from the bi-LSTM layers following noise are separately input into two bi-LSTM layers. We further concatenate these two outputs and feed them to a group of three bi-LSTM layers with latent dimension 128, 32 or 20 for further feature extraction. To create a binary mask in selection of mutation positions we build two dense layers to reduce the last dimension (the latent dimension) to 1, followed by a binary activation function that converts output data into a binary value of 0 or 1 (a value above threshold 0.5 is output as 1, otherwise as 0), where the number 1 represents selection of mutation position while 0 means that there is no need to mutate.

We then select features and output (denoted as numbers in Fig. 2a,b) from upstream and input them to the mutation block, which stimulates mutations based on the sequence. Figure 2b shows the architecture of a mutation block: a LSTM layer takes the concatenated input from the previous dense layer 1 and bi-LSTM layer 3, followed by another bi-LSTM layer with latent dimensions of 64 and 32, respectively, for further feature extraction. The dense layer then reduces the last dimension (latent dimension) to 20 using a softmax activation function. The output is a subset of base positions derived by the binary mask, and the new value is replaced at the determined mutation position at the original sequence.

We designed two discriminators, discriminator A and discriminator B, for different training purposes. The architecture of the protein complex binding-affinity change predictor and discriminator A are shown in Fig. 2c. The model takes three main types of input data: protein sequence, mutated sequence and auxiliary information. Both protein sequence and auxiliary information are the same as those for the protein sequence mutation generator described above. The mutated sequence is the sequence corresponding to the protein sequence but with mutations replacing some of its amino acids. The model calculates differences between the original and mutated sequences by subtracting their one-hot vector representing amino acids, to highlight the mutation. The original protein sequence, the sequence difference and then the auxiliary information are fed to bi-LSTM layers to encode them into vectors.

Two connected convolutional neural network (CNN) blocks take these vectors separately for extraction of features within each input vector. The binding-affinity change predictor is constructed with 32 and 64 filters in the first and second blocks, respectively, while discriminator A has 64 and 128, both with a kernel size of 5. These convolved features are concatenated and further input to another group of four CNN blocks for integrated feature extraction, with 32, 64, 128 and 256 filters for the binding-affinity change predictor and 64, 128, 256 and 512 for discriminator A, both again having a kernel size of 5. A global Max Pooling layer is then applied followed by four fully connected dense layers with neuron units 128, 64, 8 and 1. Rectified linear unit (ReLU) activation is then applied on the first two dense layers and LeakyReLU on the third, with the alpha-value set as 0.2. For the binding-affinity change predictor, a linear activation is applied at the last step of the model while for discriminator A no activation is placed. The architecture of the CNN blocks (Fig. 2d) includes a convolution two-dimensional layer, a batch normalization layer, a ReLU activation, a two-dimensional Max Pooling and a dropout layer in sequential order.

Figure 2e illustrates the architecture of discriminator B. It is composed of a group of four dense layers each having 16, 16, 8 or 1 units and with activation function ReLU, except for the last layer where a sigmoid function is set.

DeepDirect training framework and parameter settings

At stage A we use the original protein sequences and their mutated sequences for training. We first mutate the sequences via the mutation generator, label them as 'fake' and subsequently label the mutations from the training database as 'real'. We train discriminator A for five rounds and update the weights through back propagation. To prevent discriminator A from determining the sequence without generating mutations as real, we train the discriminator for three more steps to label the non-mutated sequence as fake and mutants from the training database as real. Weight clipping is applied at all steps for training of discriminator A to ensure training smoothness, then we freeze the weights of discriminator A and train the mutation generator for one step. We train discriminator A more than the mutation generator to help produce a reliable gradient. At stage B we first specify a mutation direction (in the direction of increase/decrease of binding free energies) and then we mutate the sequence via the mutation generator. Together with the original sequence, the generated mutant is then input to the pretrained binding-affinity change predictor to predict a free energy change score. We use ΔE_{G} to represent free energy changes from the mutation generator. Hence, in the circumstance of 'decrease' specified for mutation direction, the mutated data are labelled as real if $\Delta E_{\mathrm{G}} < 0$ or as fake if ≥ 0. A pseudo-energy value is then generated accordingly, sampled from uniform distribution $U(0, 2)$ and labelled as fake if $E_{\mathrm{G}} > 0$, or from $U(-2, 0)$, and labelled as real if $E_{\mathrm{G}} \geq 0$. If increase is specified for the direction, the mutated data are labelled as real if $\Delta E_{\mathrm{G}} > 0$ or as fake if ≤ 0. Subsequently a pseudo-energy value is sampled from $U(-2, 0)$ and labelled as fake if $E_{\mathrm{G}} > 0$, or sampled from $U(0, 2)$ and labelled as real if $E_{\mathrm{G}} \leq 0$. Similar to the process at stage A, we train discriminator B using this set of real and fake data for five steps with weight clipping on parameter updating. For the mutation generator we start from the trained parameter in stage A, set the generated mutant as fake and train the model without the weight-clipping restriction.

We train the protein complex binding-affinity change predictor separately and integrate it into the DeepDirect framework for training of the whole model. For training of the binding-affinity change predictor we first extract features from the original sequence, mutant, change in binding free energy and related auxiliary data. We split all data into 'train', 'validation' and 'test' at a ratio 0.7:0.15:0.15. We then randomly choose a set of training data and feed them into the predictor for training so that the model will not be overfitted to mutations specific to any typical complex.

Different loss functions are set at different stages in the training of DeepDirect. The loss function of discriminator A is designed similarly to Wasserstein loss, L_{DA}, as

$$L_{\mathrm{DA}} = \alpha \times (\overline{\text{real}} - \overline{\text{fake}}), \tag{3}$$

which is the difference in average scores (across a minibatch of samples) obtained from discriminator A between the real $\overline{\text{real}}$ and fake $\overline{\text{fake}}$ data, with α as a scalar set as 10^4. Such a loss encourages the real data to be scored lower and the fake data higher, to clearly separate real and fake training data. The loss for the mutation generator at stage A, L_{MA}, is defined as

$$L_{\mathrm{MA}} = \beta \times (\overline{\text{fake}} - \text{penalty}/\gamma) \tag{4}$$

where

$$\text{Penalty} = c_1 \times (\mathrm{SR} - R)^4 + c_2 \times (\mathrm{SR} - R) + c_3 \mathrm{SR}^2 + c_4 \tag{5}$$

and

$$\mathrm{SR} = (L - M)/L. \tag{6}$$

Here we introduce a penalty item for control of mutation rate. The penalty is calculated as equation (3) under two main parameters, R and SR; R is the mutation range, which we set at 0.8. This penalty is expected to ensure that L_{MA} decreases when R approaches our set value, keeping all other factors the same. The other parameter, SR, controls the similarity ratio, defined as the number of amino acids not mutated (sequence length L − number of mutated amino acids M) divided by sequence length L; equation (4)). We apply scalars on both penalty item γ and the entire function β set as 5 and −50, respectively, during training. Symbols c_1, c_2, c_3 and c_4 represent the coefficients of the penalty function, set as 4, 1, 1 and 0.2, respectively. We use a binary cross-entropy as the loss function for both the mutation generator and discriminator B at training stage B, L_{B}, defined as

$$L_{\mathrm{B}} = -\frac{1}{N}\sum_{i=1}^{N} y_i \times \log\left(p\left(y_i\right)\right) + (1 - y_i) \times \log\left(1 - p\left(y_i\right)\right), \tag{7}$$

where y_i is the label for the given training data, $p(y_i)$ the predicted probability of the data belonging to label y_i and N the number of training samples. We use loss of mean absolute error as the loss function for the protein complex binding-affinity change predictor L_{BAP}:

$$L_{\mathrm{BAP}} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|, \tag{8}$$

where y_i is the actual score, $\hat{y}_i$ the predicted score and N the number of samples.

The training loss of the mutation generator and discriminator in the final stage was 0.6, while the loss of the mutation generator was 0.7. Loss for the protein complex binding-affinity change predictor—mean absolute error loss—was 1.82 on the test data.

The specific settings used for training of DeepDirect model$_{\mathrm{inc}}$ and model$_{\mathrm{dec}}$ in the two sequential stages are as follows. At training stage A for discriminator A, kernel size in the CNN block was set as 5. For the first- and second-tier CNN blocks for the reference sequence, sequence difference and auxiliary information we set a filter number of 64 or 128, respectively. Neuron numbers in dense layers 1, 2, 3 and 4 were set as 128, 64, 8 and 1. respectively. Two ReLUs and one LeakyReLU with an alpha-value of 0.2 were set as the activation functions after the first three dense layers. For the mutation generator, the first-tier bi-LSTM layers after the layer of protein sequence, auxiliary information and noise all had 128 latent dimensions, and likewise for the second-tier bi-LSTM layers after the two concatenation layers. The third tier had 128, 32 and 20 latent dimensions. Both dense layers had a neuron number of 16. Within the mutation generator, the first two LSTM layers had 64 and 32 latent dimensions and the following dense layer had 20 neurons with softmax activation. The learning rate for both the discriminator and mutation generator was set at 0.000003, with Adam as optimizer. 'Clip for gradient' was set as 0.1. The ratio of updating the weights of the discriminator versus training for the discriminator using the unchanged reference sequence versus the mutation generator was set at 5:1:1.

At training stage B for discriminator B, neuron numbers in the four dense layers were 16, 16, 8 and 1. Activation functions of the three ReLUs and one Sigmoid were placed after each layer in sequential order. The mutation generator was built as in training stage 1. The learning rate for the discriminator was set at either 0.000008 or 0.000005, and learning rate for the mutation generator was set as 0.000008 and 0.00001 for model$_{\mathrm{inc}}$ and model$_{\mathrm{dec}}$, respectively. Clip for gradient was set at 0.1. We set the ratio for updating the weights of the discriminator versus mutation generator at 5:1.

For the binding-affinity predictor we set the filter number as 32 and 64, respectively, for the first and second CNN blocks for the reference sequence, sequence difference and auxiliary information. A linear activation function was placed after the final dense layer. The remainder were assembled as for discriminator A. Adam was used as the optimizer for training, with a learning rate of 0.0001.

Docking and random mutation process

Docking was performed with the HDOCK server, between PDB entry 7JJI (SARS-CoV-2 3Q-2P full-length prefusion spike trimer) and Native Human 1R42 (angiotensin-converting enzyme-related carboxypeptidase (ACE2)). We selected the top-predicted model having a HDOCK docking score of −261 and a root-mean-square deviation score of 343.62. The amino acid 3D coordinates of the docked protein complex were extracted as the input to DeepDirect to generate mutations. In the generation of random mutations, we first randomly chose mutation positions in a protein sequence restricted by a mutation rate that we set at 5, 10 or 20% in this work. At each determined mutation site the amino acid is randomly substituted with a different amino acid. Such a mutation pipeline was applied to generate 500 mutated sequences under each mutation rate; 3D structures of protein complexes were visualized by ChimeraX software[47].

Evolution analysis of the SARS-CoV-2 Omicron strain

The PDB file of the SARS-CoV-2 Omicron Variant SPIKE trimer complexed with ACE2 was downloaded under accession no. 7WPA. The relevant information was extracted and input to DeepDirect model$_{\text{inc}}$ 500 times iteratively, with a batch size of 500. Mutations on spike protein chain A were extracted to construct the mutated spike trimer and used in the analysis. Random mutations were generated as for the previous random mutation generation steps. Sequence Graph Transform was applied via the SGT package with parameter kappa set at 5. PCA and DBSCAN were applied via sklearn. DBSCAN had the parameter eps set as 0.015, determined by k-NN distance (Supplementary Fig 2a). Package Numpy and Seaborn were used for data processing and visualization.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The AB-bind database was downloaded from https://github.com/sarahsirin/AB-Bind-Database (ref. 43). The SKEMPI v.2.0 database was downloaded from https://life.bsc.es/pid/skempi2/database/index (ref. 44). The PDB structures of 7JII, 7WPA and 1R42 were downloaded from https://www.rcsb.org/structure/7JII, https://www.rcsb.org/structure/7WPA and https://www.rcsb.org/structure/1r42, respectively.

Code availability

The source code for DeepDirect is available at https://github.com/tianlt/Deepdirect (ref. 48), where some analysis results are also provided.

References

1. Nero, T. L., Morton, C. J., Holien, J. K., Wielens, J. & Parker, M. W. Oncogenic protein interfaces: small molecules, big challenges. *Nat. Rev. Cancer* **14**, 248–262 (2014).
2. Levitt, M. & Warshel, A. Computer simulation of protein folding. *Nature* **253**, 694–698 (1975).
3. Bianchi, F. et al. Steric exclusion and protein conformation determine the localization of plasma membrane transporters. *Nat. Commun.* **9**, 501 (2018).
4. Doerr, A. Tracking protein conformation in live cells. *Nat. Methods* **18**, 1451 (2021).
5. Chen, S.-J. et al. Protein folds vs. protein folding: differing questions, different challenges. *Proc. Natl Acad. Sci. USA* **120**, e2214423119 (2023).
6. Tsay, Y.-F. How to switch affinity. *Nature* **507**, 44–45 (2014).
7. Ozono, S. et al. Sars-cov-2 d614g spike mutation increases entry efficiency with enhanced ace2-binding affinity. *Nat. Commun.* **12**, 848 (2021).
8. Chen, D. et al. Regulation of protein-ligand binding affinity by hydrogen bond pairing. *Sci. Adv.* **2**, e1501240 (2016).
9. Bennett, N. R. et al. Improving de novo protein binder design with deep learning. *Nat. Commun.* **14**, 2625 (2023).
10. Wu, F., Jing, X., Luo, X. & Xu, J. Improving protein structure prediction using templates and sequence embedding. *Bioinformatics* **39**, btac723 (2023).
11. Liu, X., Luo, Y., Li, P., Song, S. & Peng, J. Deep geometric representations for modeling effects of mutations on protein-protein binding affinity. *PLoS Comput. Biol.* **17**, e1009284 (2021).
12. Xiong, P., Zhang, C., Zheng, W. & Zhang, Y. Bindprofx: assessing mutation-induced binding affinity change by protein interface profiles with pseudo-counts. *J. Mol. Biol.* **429**, 426–434 (2017).
13. Joughin, B. A., Green, D. F. & Tidor, B. Action-at-a-distance interactions enhance protein binding affinity. *Protein Sci.* **14**, 1363–1369 (2005).
14. Moal, I. H., Agius, R. & Bates, P. A. Protein–protein binding affinity prediction on a diverse set of structures. *Bioinformatics* **27**, 3002–3009 (2011).
15. Kundrotas, P. J., Zhu, Z. & Vakser, I. A. Gwidd: genome-wide protein docking database. *Nucleic Acids Res.* **38**, D513–D517 (2010).
16. Moulana, A. et al. The landscape of antibody binding affinity in sars-cov-2 omicron ba. 1 evolution. *eLife* **12**, e83442 (2023).
17. Co, M. S. et al. Genetically engineered deglycosylation of the variable domain increases the affinity of an anti-cd33 monoclonal antibody. *Mol. Immunol.* **30**, 1361–1367 (1993).
18. Maute, R. L. et al. Engineering high-affinity pd-1 variants for optimized immunotherapy and immuno-pet imaging. *Proc. Natl Acad. Sci. USA* **112**, E6506–E6514 (2015).
19. Yugandhar, K. & Gromiha, M. M. Protein–protein binding affinity prediction from amino acid sequence. *Bioinformatics* **30**, 3583–3589 (2014).
20. Abbasi, W. A., Yaseen, A., Hassan, F. U., Andleeb, S. & Minhas, F. U. A. A. Island: in-silico proteins binding affinity prediction using sequence information. *BioData Min.* **13**, 20 (2020).
21. Öztürk, H., Özgür, A. & Ozkirimli, E. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics* **34**, i821–i829 (2018).
22. Stepniewska-Dziubinska, M. M., Zielenkiewicz, P. & Siedlecki, P. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics* **34**, 3666–3674 (2018).
23. Rifaioglu, A. S. et al. Mdeepred: novel multi-channel protein featurization for deep learning-based binding affinity prediction in drug discovery. *Bioinformatics* **37**, 693–704 (2021).
24. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
25. Tubiana, J., Schneidman-Duhovny, D. & Wolfson, H. J. Scannet: an interpretable geometric deep learning model for structure-based protein binding site prediction. *Nat. Methods* **19**, 730–739 (2022).
26. Renaud, N. et al. Deeprank: a deep learning framework for data mining 3d protein-protein interfaces. *Nat. Commun.* **12**, 7068 (2021).
27. Xu, J., Mcpartlon, M. & Li, J. Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nat. Mach. Intell.* **3**, 601–609 (2021).
28. Ward, M. D. et al. Deep learning the structural determinants of protein biochemical properties by comparing structural ensembles with diffnets. *Nat. Commun.* **12**, 3023 (2021).
29. Baek, M. & Baker, D. Deep learning and protein structure modeling. *Nat. Methods* **19**, 13–14 (2022).
30. Van Kempen, M. et al. Fast and accurate protein structure search with foldseek. *Nat. Biotechnol.* **42**, 243–246 (2024).
31. Senior, A. W. et al. Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).
32. Dauparas, J. et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science* **378**, 49–56 (2022).
33. Wicky, B. et al. Hallucinating symmetric protein assemblies. *Science* **378**, 56–61 (2022).
34. Yeh, A. H.-W. et al. De novo design of luciferases using deep learning. *Nature* **614**, 774–780 (2023).
35. Motmaen, A. et al. Peptide-binding specificity prediction using fine-tuned protein structure prediction networks. *Proc. Natl Acad. Sci. USA* **120**, e2216697120 (2023).
36. McPartlon, M. & Xu, J. An end-to-end deep learning method for protein side-chain packing and inverse folding. *Proc. Natl Acad. Sci. USA* **120**, e2216438120 (2023).

37. Goodfellow, I. et al. Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2020).
38. Mirza, M. & Osindero, S. Conditional generative adversarial nets. Preprint at https://doi.org/10.48550/arXiv.1411.1784 (2014).
39. Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint at https://doi.org/10.48550/arXiv.1511.06434 (2015).
40. Nowozin, S., Cseke, B. & Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. *Adv Neural Inf Process Syst.* **29** (2016).
41. Huang, A.-D., Zhong, Z., Wu, W. & Guo, Y.-X. An artificial neural network-based electrothermal model for GaN HEMTs with dynamic trapping effects consideration. *IEEE Trans. Microw. Theory Tech.* **64**, 2519–2528 (2016).
42. Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning* (eds Precup, D. & Teh, Y. W) 214–223 (PMLR, 2017).
43. Sirin, S., Apgar, J. R., Bennett, E. M. & Keating, A. E. Ab-bind: antibody binding mutational database for computational affinity predictions. *Protein Sci.* **25**, 393–409 (2016).
44. Jankauskaitė, J., Jiménez-García, B., Dapkūnas, J., Fernández-Recio, J. & Moal, I. H. Skempi 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics* **35**, 462–469 (2019).
45. Ranjan, C., Ebrahimi, S. & Paynabar, K. Sequence graph transform (sgt): a feature embedding function for sequence data mining. *Data Min. Knowl. Discov.* **36**, 668–708 (2022).
46. Ester, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD* **96**, 226–231 (1996).
47. Pettersen, E. F. et al. Ucsf chimerax: structure visualization for researchers, educators, and developers. *Protein Sci.* **30**, 70–82 (2021).
48. Lan, T. DeepDirect. *Zenodo* https://doi.org/10.5281/zenodo.10004503 (2023).

Acknowledgements

This work was supported in part by Australia Research Council Discovery Project (grant no. DP180100120 to J.L.) and start-up funding grants for new staff (J.L.) at the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences.

Author contributions

T. Lan developed the model, performed analysis and wrote the manuscript. T. Lan and J.L. conceptualized the project and designed the analysis steps. J.L. revised the manuscript and supervised the work. T. Lan and P.P. performed the evaluation experiments and wrote software documentation. T. Lan and S.S. created visualizations of the results. S.S., P.P., G.H., T. Liu and Y.P. provided suggestions and technical support on the project. All authors discussed the results and commented on the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-024-00803-z.

Correspondence and requests for materials should be addressed to Jinyan Li.

Peer review information *Nature Machine Intelligence* thanks Dong Xu and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

[1]School of Computer Science, Faculty of Engineering and IT, University of Technology Sydney, Ultimo, New South Wales, Australia. [2]Data Science Institute, Faculty of Engineering and IT, University of Technology Sydney, Ultimo, New South Wales, Australia. [3]Natexl, Ultimo, New South Wales, Australia. [4]School of Biomedical Engineering, Faculty of Engineering and IT, University of Technology Sydney, Ultimo, New South Wales, Australia. [5]Children's Cancer Institute Australia and School of Women's and Children's Health, University of New South Wales, Sydney, New South Wales, Australia. [6]Academy of Medical Sciences, Zhengzhou University, and Translational Research Institute, Henan Provincial People's Hospital, Zhengzhou, China. [7]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. [8]These authors contributed equally: Shuquan Su, Pengyao Ping. ✉e-mail: jinyan.li@siat.ac.cn

nature portfolio

Corresponding author(s): Jinyan Li

Last updated by author(s): Jan 30, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
□	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	□	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	□	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	□	A description of all covariates tested
☒	□	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
□	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	□	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	□	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	□	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	□	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	no software was used
Data analysis	numpy 1.19.5 pandas 1.3.0 seaborn 0.11.2 Chimerax 1.7 https://github.com/tianlt/Deepdirect

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

AB-Bind database was downloaded from https://github.com/sarahsirin/AB-Bind-Databas
SKEMPI2.0 database was downloaded from hhttps://life.bsc.es/pid/skempi2/database/index
PDB structure of 7JII, 7WPA, 1R42 were downloaded from https://www.rcsb.org/structure/7JII, https://www.rcsb.org/structure/7WPA and https://www.rcsb.org/structure/1r42, respectively.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	This study did not involve human research participants.
Population characteristics	This study did not involve human research participants.
Recruitment	This study did not involve human research participants.
Ethics oversight	This study did not involve human research participants.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Sample size n = 500 for each condition
Data exclusions	No data exclusions
Replication	We include large sample sizes n = 500 for each condition.
Randomization	Not relevant as all experimental conditions starting from the same initial sequence.
Blinding	Not relevant as all experimental conditions starting from the same initial sequence.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	□ Antibodies
☒	□ Eukaryotic cell lines
☒	□ Palaeontology and archaeology
☒	□ Animals and other organisms
☒	□ Clinical data
☒	□ Dual use research of concern

Methods

n/a	Involved in the study
☒	□ ChIP-seq
☒	□ Flow cytometry
☒	□ MRI-based neuroimaging