Finding Antibodies in Cryo-EM densities with CrAI

Vincent Mallet [1], **Chiara Rapisarda** [2], **Hervé Minoux** [2], and **Maks Ovsjanikov**[1]

[1]LIX, Ecole Polytechnique, IPP Paris
[2]Sanofi
maks@lix.polytechnique.fr

Abstract

Therapeutic antibodies have emerged as a prominent class of new drugs due to their high specificity and their ability to bind to several protein targets. Once an initial antibody has been identified, an optimization of this hit compound follows based on the 3D structure, when available. Cryo-EM is currently the most efficient method to obtain such structures, supported by well-established methods that can transform raw data into a potentially noisy 3D map. These maps need to be further interpreted by inferring the number, position and structure of antibodies and other proteins that might be present. Unfortunately, existing automated methods addressing this last step have a limited accuracy and usually require additional inputs, high resolution maps, and exhibit long running times.

We propose the first fully automatic and efficient method dedicated to finding antibodies in cryo-EM densities: CrAI. This machine learning approach leverages the conserved structure of antibodies and exploits a dedicated novel database that we built to solve this problem. Running a prediction takes only a few seconds, instead of hours, and requires nothing but the cryo-EM density, seamlessly integrating in automated analysis pipelines. Our method is able to find the location of both Fabs and VHHs, at resolutions up to 10Å and is significantly more reliable than existing methods. It also provides an accurate estimation of the antibodies' pose, even in challenging examples such as Fab binding to VHHs and vice-versa. We make our method available as a ChimeraX[44] bundle. [1]

Since the first monoclonal antibody entered the clinic in 1986 [19], antibody-based therapeutics have made considerable progress. With over a hundred compounds approved and forty just in the last three years [33, 55], the use of antibodies currently appears as one of the most promising approaches for designing new treatments for patients. Antibody-based therapeutics rely on the identification of antibodies that can bind to a target molecule (antigen) with high specificity through their tips called the Complementarity Determining Regions (CDRs). Monoclonal antibodies (mAb) are the most widely used antibodies. They are composed of two domains called the Antigen Binding Fragment (Fab) and a central stalk that binds to the Fc receptor. More recently, antibody fragments consisting of a single variable antibody domain, engineered from heavy-chain antibodies generated by camelids [1] (nAbs or VHHs) - have attracted interest as an alternative to mAbs [17, 30]. Whether for mAbs or VHHs, initial hits are typically found using immunization [29, 35] or phage display [41, 54]. Before entering clinical studies, initial hits need to be optimized with regards to several properties including their efficacy, manufacturability and safety. Among those properties, the specific binding to the antigen is a critical objective scrutinized from the early phases of the process until drug candidate selection. Binding optimization relies on obtaining the structure of the initial hit since the knowledge of the atomic coordinates at the contact points between the Ab and its target improves the understanding of its mode of action and guides the optimization of the binding affinity [12].

[1]github.com/Sanofi-Public/crai and cxtoolshed.rbvi.ucsf.edu/apps/chimeraxcrai

Cryogenic Electron Microscopy (cryo-EM) has become the most common way to experimentally obtain protein structures of therapeutic antibodies bound to their target. In cryo-EM, the target protein is embedded in ice and exposed to an electron beam, resulting in raw noisy images of individual particles. These raw 2D images are then aligned and transformed into a 3D electron probability density [48, 52] from which atomic coordinates are inferred. Recent advances in data collection hardware and software, along with improved data processing pipelines have increased data output. As a result, more academic labs and global pharmaceutical industry have adopted the technology[42]. Lately, a new data collection workflow has been shown to produce 3-4Å structures of a pharmaceutically relevant target protein with 1 hour of instrument time, thus allowing the theoretical resolution of 24 structures a day [15]. However, this raw data needs to be processed from micrographs to molecular structures. While data collection is rapid, current data *processing pipelines* rely on significant manual intervention for simple tasks and decisions, and ultimately take days and weeks to complete.

The recent rise of artificial intelligence-based methods for protein structure prediction holds promise but cannot yet accurately model of the interaction between antibodies and their epitope. Current methods fail to accurately model the diverse CDR loops that are critical for antigen binding [27] calling for significantly more structural data. Unfortunately, the reliance of existing methods on manual intervention significantly hinders cryo-EM density analysis at scale, required for such understanding.

As automation has allowed X-ray crystallography to become a key technique in the structure-based drug discovery pipeline [4], so it should for cryo-EM, making it cheaper and faster, and freeing the time of researchers from button pressing tasks, to structure interpretation and drug engineering. The process of automation is being accelerated by the application of machine learning to the different stages of the pipeline, from data acquisition [5, 21], to preprocessing of micrographs [51], particle picking [2, 58, 59, 60, 61], 2D class selection [34] and 3D heterogeneity deconvolution [40, 62].

Unfortunately, the last step of attribution of the map (fitting of atomic coordinates of a protein into the density), remains a tedious analysis bottleneck. It is still a largely manual process typically done using ChimeraX [43], followed by the local optimization of atomic coordinates using Coot [20]. Some techniques have been developed to help automate this process [37], although with limited accuracy. Furthermore, the methodological challenges associated with this problem have so far impeded the automation of this step using existing machine learning approaches. Specifically, the data is noisy and heterogeneous and the output is high dimensional, making off-the-shelf computer vision methods irrelevant. Tools based on machine learning to trace the sequence in the density were recently developed with good results [28, 46]. They are, however, limited to resolutions better than 4Å and can exhibit prohibitive running times.

In the context of using cryo-EM for optimization of therapeutic Abs, we aim to address the problem of finding Abs (Fabs and VHHs) in cryo-EM maps. To achieve this, we propose `CrAI`, the first fully automatic and efficient machine learning based approach that is applicable at all resolutions better than 10Å without any additional inputs beyond the density. To develop our solution, we introduce a customized deep learning technique, which takes into account and exploits the structural properties of this problem setting. In particular, we leverage the conserved structure of Abs as a prior information[14] to formulate our problem as a special instance of 3D object detection [49, 64]. We gather a novel database of aligned Ab structures and densities, and use it to train a model with a custom loss that involves optimal transport supervision. We test our tool on a set of 215 maps of various resolutions, containing 374 Fabs and 86 VHHs. We successfully find Abs in over 90% of systems, outperforming existing methods by a margin of 25%, while exhibiting **one thousand times** speedup. We make our tool available as a ChimeraX bundle to facilitate adoption.

Results

Overview

`CrAI` detects antibodies in cryo-EM maps using a customized deep learning-based technique, trained on our curated dataset comprised of 1430 cryo-EM maps containing Fabs and VHHs. When designing the model, we introduce a custom representation of the structure of antibodies to facilitate the learning process. Specifically, we leverage the *conserved nature* of antibodies to approximate the detailed structure of the output by its *position* and *orientation*. Moreover, we parameterize the orientation in a

biologically relevant way that favors the prediction of the CDRs location. Finally, the list of such representations of antibodies for a system is transformed into a grid overlaid over the cryo-EM map, such that the position of an antibody is encoded as an offset from a grid cell. The encoding of the output is shown in Figure 1A.

We use an architecture that consists of a 3D UNet trained to predict this output representation from the input map using a custom loss with two components. The first one enforces the prediction of the location of grid cells containing an antibody, and relies on optimal transport to take into account the geometric aspect of the prediction. The other part controls the prediction of the pose and the discrimination between Fabs and VHHs. Once the model makes a prediction, persistence diagrams [6, 7] are used to select the relevant results which are transformed into a PDB file. Remarkably, this procedure does not require anything beyond the input density. Our pipeline is illustrated in Figure 1.

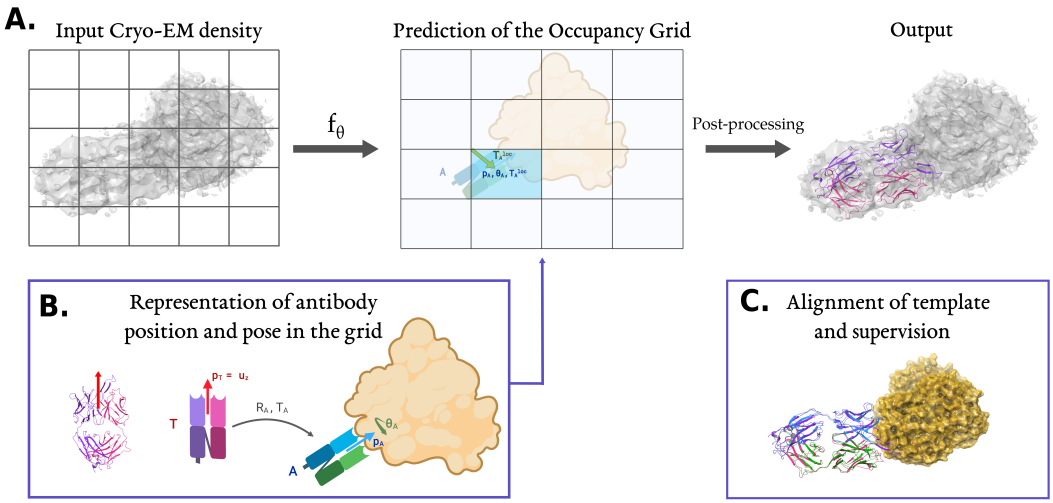


Figure 1: **A.** `CrAI` predicts an occupancy grid that represents antibodies found in an input density. The prediction of this occupancy grid can be post-processed into a PDB containing the predicted antibodies' structures. **B.** The atomic structure of our template (pdb `7lo8`) is displayed next to its cartoon, with $\vec{u}_z$ shown in red. We compute optimal alignments $R^*_{\mathbf{A}}, T^*_{\mathbf{A}}$ of our template onto Abs and decompose the rotation into $R^*_{\mathbf{A}} = (\vec{p}_{\mathbf{A}}, \theta_{\mathbf{A}})$. We decompose the translations into a position in a grid $T^{cell}_{\mathbf{A}}$ and an offset from the grid corner $T^{loc}_{\mathbf{A}}$. We thus obtain a grid with zero values except for cells containing an Ab, in those cells, we have $\vec{p}_{\mathbf{A}}$, $\theta_{\mathbf{A}}$ and $T^{loc}_{\mathbf{A}}$. **C.** Example alignment of our template (*red, purple*) with the experimental structure of system `6bf9`: antigen (*orange*) and Fab (*blue, green*). As can be seen, our template aligns well to other antibodies (RMSD = 1.8Å).

Baselines and Metrics

To the best of our knowledge, no tool enables predicting the position of antibodies solely from a density map. We benchmark against `dock_in_map` [37], a tool that takes the density along with the known atomic protein structures that need to be docked in the map. Considering that at test time, one does not have access to the ground truth structure, we ran `dock_in_map` with a fixed template Fab or Fv. Additionally, we ran `dock_in_map` with the actual structure giving us an upper bound of its performance.

Independent models were trained on the training sets obtained with the `random` and `sorted` splits (see Methods Section). We make inference of those models on their respective test sets, providing the network with the number of Abs to find (denoted as `num`) or relying on automatic thresholding to infer this number (`thresh`). For each system, a prediction method results in one or several predicted Ab positions. When evaluating different approaches, individual predicted antibody positions need to be matched with an actual antibody position. This matching process is accomplished using the Hungarian algorithm [36] on the distance between the center of mass of predicted and actual antibodies.

The distribution of distances between centers of mass of the predictions and experimental structures shows that we outperform `dock_in_map`, even in the idealized scenario that uses the ground truth structure (see Supplementary Figure S.2). In the following, we will compare to this idealized and more challenging scenario as our baseline. Moreover, we observe that the predicted distances are bimodal: a first peak corresponds to systems predicted successfully and another spread mode corresponds to failed prediction. Hence, we will report our results in terms of **hit rates**: a prediction is deemed as a hit if it is closer than 10Å to the ground truth. As can be seen in the histogram, performance is not very sensitive to the choice of this threshold. We report our results for individual Abs (`ab`) as well as aggregated by systems (`sys`) so that a system with many Fabs does not influence the results significantly.

`CrAI` accurately finds Abs

Our key results of Fab detection are presented in Table 1. In the first two rows, note that `CrAI` drastically outperforms `dock_in_map` in terms of hit rates. This holds true in all settings for an overall hit rate going from approximately 69% to 95%. In terms of distance between the centers of mass prediction and the ground truth, both methods give results below 2.5Å which is close to optimal considering the map resolutions. Given a successful hit, `dock_in_map` appears to be a bit more precise, which is expected as it uses the ground truth structure instead of a template. Moreover, distances are computed only over successful systems and thus include *25% more systems in the `CrAI` column*. One could first rapidly screen a map with a high hit rate using `CrAI`, then precisely refine results in a density neighborhood using ChimeraX fast local refinement tool, *fitmap*. This possibility is enabled in our ChimeraX bundle.

Method	Hit rate						Hit distance	Runtime	
	random		sorted		mean				
	sys	ab	sys	ab	sys	ab		CPU	GPU
`dock_in_map`	66.6	61.9	71.3	66.0	69.0	64.0	**0.73A**	883s	-
`CrAI`	**97.3**	**96.7**	**96.9**	**95.7**	**97.1**	**96.2**	1.73A	**1.9s**	**0.47s**
`CrAI thresh`	95.9	95.2	92.7	92.0	94.3	93.6	1.71A	**1.9s**	**0.47s**

Table 1: Fab detection performance of the performance of the benchmark tool `dock_in_map` and of `CrAI`, provided with the ground truth number of objects (`num`) or not (`thresh`). We report the Hit Rate (defined as the number of prediction below a threshold of 10Å) on different data splits (`random` or `sorted`), aggregated by systems (`sys`) or not (`ab`). We additionally report the mean distance of these successful predictions, and the mean runtime of the tool.

`CrAI` can be used with unknown number of Fabs We now evaluate `CrAI` in the scenario where it also automatically estimates the *number* of Abs, and thus the network is given *only the density* without any additional information. As shown in the second and third lines of Table 1, we retain most of the performance even in this more challenging, and more realistic setting. We find this remarkable, since the densities are highly heterogeneous and contain between one and six Abs. This is to compare to the result of `dock_in_map` that is additionally given the *experimental structures of all antibodies* that are to be found in the map. In contrast, the fact that our approach can adapt to different scenarios strongly highlights its effectiveness and flexibility, and sets it apart from *any* existing method in terms of its ability to detect antibodies in Cryo-EM densities in a fully automatic manner. Furthermore, as highlighted below, our approach is also orders of magnitude faster than the baselines, making it the first truly scalable and practical method for efficiently detecting Fabs in Cryo-EM densities.

`CrAI` runs fast, at all resolutions The average runtime of `dock_in_map` on our validation set is a prohibitive 883s/system. This mean is optimistic because we stopped a few systems after 5 hours of computations (letting them run longer would increase the mean). In comparison, our tool runs in **0.47s/system,** i.e., more than a thousand times faster when using a GPU (A40). Even when using only one CPU, our tool runs in 1.9s/system, four hundred times faster than `dock_in_map` and fast enough for this computational step to integrate seamlessly in an analysis pipeline. This stems from the complexity of our algorithm, that is linear with regards to the grid size.

Supplementary Figure B.2 shows how our performance depends on resolution of the input map, over all of our data sets. Contrary to `dock_in_map`, we do not see a correlation and thus claim that `CrAI` is *robust to low resolution*. This is a novel result as machine learning methods for tracing such as ModelAngelo [28] only work for resolutions better than 4Å.

`CrAI` **finds VHHs** We now turn to the prediction of VHHs. They are more challenging for a machine learning model, because we have less training data (278 vs 722 training examples). Moreover, the binding modes of VHHs are less canonical which can affect the decomposition of our rotation. Finally, VHHs are smaller and tend to be more buried into the density - as opposed to Fabs often sticking out. Our results are presented in Table 2.

Method	Hit rate						Hit distance
	random		sorted		mean		
	sys	ab	sys	ab	sys	ab	
`dock_in_map`	77.8	68.8	90.5	89.1	84.2	79.0	**0.5A**
`CrAI num`	**100**	**100**	**90.9**	**90.6**	**95.5**	**95.3**	1.66A
`CrAI thresh`	99.1	98.7	84.1	81.2	91.6	90.0	1.57A

Table 2: Performance at detecting VHHs. The convention used is the same as above.

We achieve a positive result with a 15% hit rate boost over `dock_in_map` overall, and 9% in the `thresh` setting. Beyond the intrinsic difficulties outlined above, we underline two additional factors to explain why the performance gap is smaller than for Fabs. On average, `dock_in_map` performs much better for VHHs than for Fabs, which could stem from their better resolutions. Moreover, on the `sorted` split, it outperforms itself (performance over the whole data is similar to *random*), which can be explained by the variance of estimators on small sized test sets. Nevertheless, we emphasize that `CrAI` still outperforms `dock_in_map` in terms of the number of successful detections, without relying on any additional side (ground truth) input and being orders of magnitude faster.

Ablation study To assess the relevance of different design choices of our approach, we retrain several models without specific individual features in our design. Since this means training a new model every time, we only perform this analysis in the Fab `random` split setting. We report the performance in the `sys` setting, but results are consistent in the `ab` setting. We try replacing Persistence Diagrams (PD) with a naive Non-Maximal Suppression (NMS) [22, 25] that amounts to zeroing predictions around local minima. We also trained our model without using the Optimal Transport (OT) component, as in done in most classical object detection approaches [24, 50]. We also tried to keep the method fixed, but to disrupt the template encoding by decomposing the rotation using u_y as the main vector. We present the results in Table 3.

Method	Hit Rate	
	`num`	`thresh`
without PD	**97.3**	95.5
without OT	88.6	0
$u_z \to u_y$	92.6	89.5
`CrAI`	**97.3**	**95.9**

Table 3: Hit rates for different ablations of our method, on the Fab `random` split, in the `sys` setting.

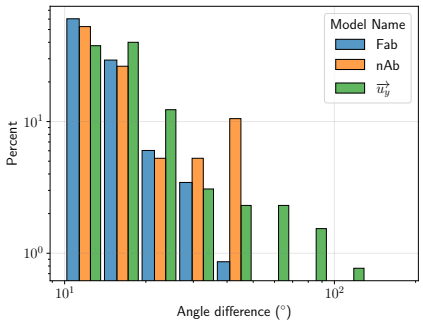


Figure 2: Histograms of the angles between the predicted and experimental $\vec{p}_{\mathbf{A}}$ vector for Fabs, VHHs and $\vec{u}_y$ model

Persistence diagrams seem to enhance results in the `thresh` setting, with a limited impact. However, when removing optimal transport, performance collapses especially on the ability to predict the

number of objects . This can be explained considering that antibodies cannot overlap (in contrast to detecting pedestrians in images for instance). Optimal transport helps to attribute a single detection to a region rather than enabling an arbitrary number of potentially overlapping/conflicting detections. Training the model with u_y instead of u_z also significantly weakens detection performance.

`CrAI` **finds meaningful poses** After validating the position of our predictions, we consider the predicted poses: are Abs in the correct orientation? Using the decomposition of rotations into predicting a vector $\vec{p}_{\mathbf{A}}$ and an angle $\theta_{\mathbf{A}}$, we can compute the angles between predicted values and actual ones. We provide histograms for the distribution of these angles in Figure 2, to show that most systems are predicted accurately. For the Fab data, the angle between $\vec{p}_{\mathbf{A}}$ and its prediction is on average of 7.8° and the one for $\theta_{\mathbf{A}}$ is 11.0°, low enough values to make the prediction almost overlap with ground truth. For the VHH data, those values are on average 7.5° and 9.7° respectively. Finally, the model trained using $\vec{u}_y$ instead of $\vec{u}_z$ has an angle error of respectively 10.2° and 11.6°, significantly higher on the vector prediction. Hence, predicting $\vec{u}_z$ is easier than $\vec{u}_y$, which validates the non-restrictive inductive bias that we introduced in our formulation.

Further analysis

Case studies Detailed evaluation results for 6 systems of the random split test set are shown in Figure 3. Those six systems were picked because of their relevance in the context of drug discovery, at resolutions ranging from 3 to 8Å, include four Fabs whose location was found by our tool consistently and correctly. `dock_in_map` on the other hand, only finds the right location in one case, predicts a shifted location on another and completely fails its prediction on the last ones. The two remaining systems, with resolutions of 3.1 and 3.8Å respectively, contain VHHs for which `CrAI` detects the location correctly and for which `dock_in_map` fails. There were no examples in which dock in map succeeded and `CrAI` failed. Of the 6 systems shown, 4 used Chimera for manual docking, one doesn't mention the docking program and one uses MDFF[56], a molecular dynamics software not commonly employed by experimental structural biologists. It is hard to evaluate ChimeraX placement in the map as the time and the success of placement isn't automatic and it is user dependent. For the lower resolution structures, chain building programs are not suitable as resolutions better than 3.5 Å are needed.

The first two examples are low resolution illustrations of the structural characterization of neutralizing antibodies, in the context of cervical cancer for the Human papillomavirus type 58 pseudovirus (HPV58) in complex with the Fab fragment of 5G9 at a resolution of 6.41Å [26](Figure 3a) and the viral infection with tetanus for the neurotoxin LC-HN domain in complex with TT110- Fab1 at a resolution of 8.00Å [47] (Figure 3b). While the low resolution of the first example does not preclude both `CrAI` and `dock_in_map` to correctly predict the position of the Fab, at 8.0 Å, only `CrAI` correctly positions the Fab. Membrane proteins are notoriously difficult to characterize structurally and are often solved at lower resolutions than soluble ones. Structural information is important to decipher the signaling and regulatory mechanisms of this class of proteins. G protein-coupled receptors (GPCRs) in Figure 3c and d play a crucial role in signaling and are one of the major target family of drugs. Three Fabs are present in the density of beta-arrestin2 (Figure 3c) in complex with a phosphopeptide corresponding to the human C5a anaphylatoxin chemotactic receptor 1 C5aR1 at a resolution of 4.40Å [39]. `CrAI` correctly finds all of them while `dock_in_map` misses to correctly predict two out of three. In Figure 3d, the class A orphan GPCR (GPR139), [63] is captured with and stabilized by Nb35 in different conformational states highlighting allosteric modulation roles of VHHs in GPCRs. In this first example with a VHH in a protein complex, `CrAI` correctly finds the location of the VHH while `dock_in_map` failed. In the fight against covid-19, a search for potent and broad-spectrum coronavirus blockers was initiated and many antibodies were shown to block the entry of several variants of SARS-CoV-2 (SARS-CoV-2, SARS-CoV-2-D614G, B.1.1.7, B.1.351, B.1.617.1, P.1, SARS-CoV, and HCoV-NL63) by binding to human ACE2, without causing severe side effects. In Figure 3e, ACE2 is found in complex with targeting monoclonal antibodies at a resolution of 3.2Å [10]. This system illustrates `CrAI` correctly finds the two Fabs in the density, while `dock_in_map` is unsuccessful. The final example in Figure 3f is the Cryo-EM structure of SPCA1a in E1-Ca-AMPPCP state subclass 3 at a resolution 3.14Å [11]. In this case, with a medium resolution, the location of the VHH Nb14 is properly found by `CrAI`, however, its orientation is not correct. When analyzing this protein complex, it appears that the binding mode of Nb14 to its antigen, the

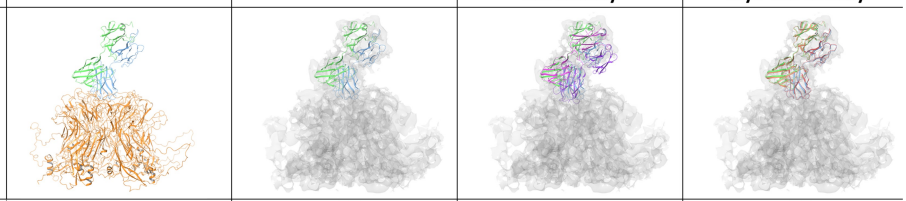
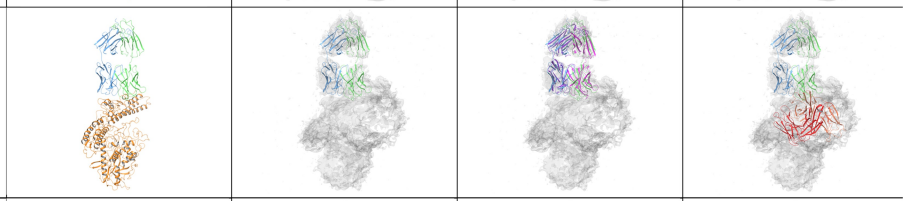
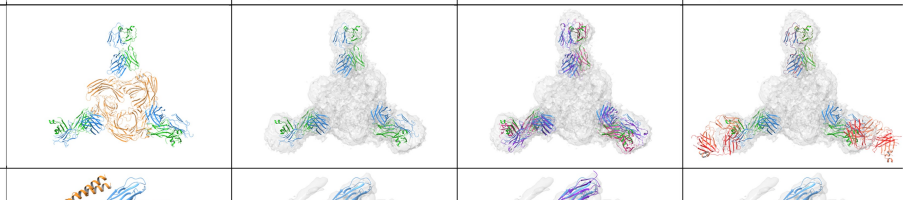
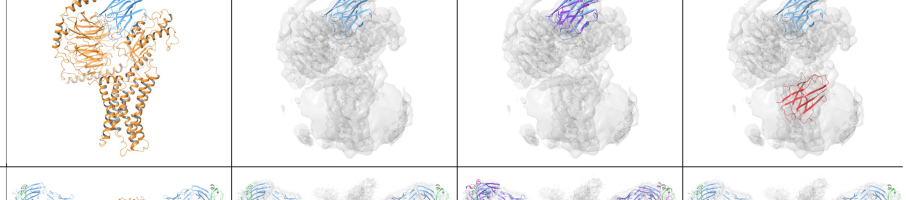
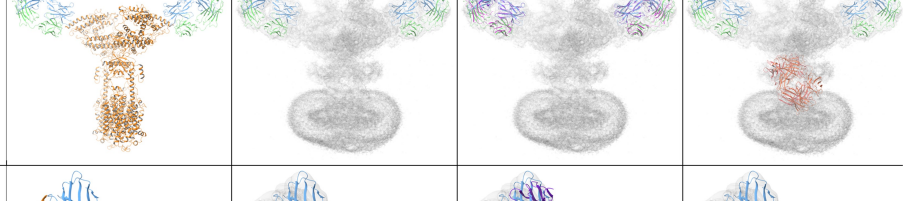
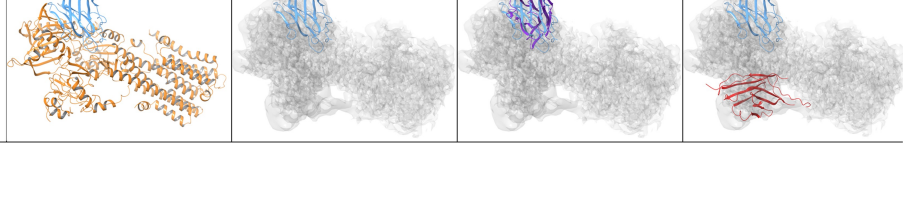
	PDB ID (resolution)	Protein complex structure	Fab/VHH structure and Cryo-EM density	Fab/VHH structure, CrAI results and Cryo-EM density	Fab/VHH structure, Dockim results and Cryo-EM density
a	7DNK (6.41 Å)				
b	7OH1 (8.00 Å)				
c	8GOO (4.40 Å)				
d	7VUJ (3.80 Å)				
e	7V61 (3.20 Å)				
f	7YAI (3.14 Å)				

Figure 3: In this Table, columns correspond to 1) PDB code and EM resolution; 2) Protein structures from the PDB (the antigen is displayed in orange, the heavy chain in blue and the light chain in green); 3) EM map and its corresponding structure; 4) Superimposition of the Fab or the VHH from the PDB structure (blue/green) with the `CrAI` results (purple); 5) Superimposition of the FAB or the VHH from the PDB structure with the `dock_in_map` results (red).
The rows correspond to six systems of our test set: (a) Human papillomavirus type 58 pseudovirus (HPV58) in complex with the Fab fragment of 5G9 at a resolution of 6.41Å, (b) Tetanus neurotoxin LC-HN domain in complex with TT110- Fab1 at a resolution of 8.00Å, (c) Structure of beta-arrestin2 in complex with a phosphopeptide corresponding to the human C5a anaphylatoxin chemotactic receptor 1 C5aR1 at a resolution of 4.40Å, (d) Cryo-EM structure of a protein complex containing a class A orphan GPCR (GPR139), stabilized by a VHH (Nb35) at a resolution of 3.80Å, (e) ACE2 -Targeting Monoclonal Antibody as Potent and Broad-Spectrum Coronavirus Blocker at a resolution of 3.2Å, (f) Cryo-EM structure of SPCA1a in E1-Ca-AMPPCP state subclass 3 at a resolution 3.14Å.

Calcium transporting ATPase type 2C member 1, is non-conventional, with the framework region of Nb14 lying on the surface of the antigen while the CDRs are not oriented toward the epitope. The prediction made by `CrAI` followed the conventional binding mode of a VHH, with the CDRs facing the epitope of the antigen. This specific example highlights a potential bias of `CrAI` in correctly predicting the orientation and the binding mode of antibodies and VHH. It can be noted that in this specific case, `dock_in_map` fails to identify the correct location of Nb14.

Failure analysis Since the number of failed systems with `CrAI` is relatively low, we visualized all *all of them* (for the `sorted` split) in Supplementary Figure S.3. After inspecting these failures, we identified two dominant sources of errors as reported in Table 4A. A visual depiction of these two error categories is shown in Figure 4B.

Error type	Count	
	`sys`	`ab`
Extra Abs	13	14
- correctly placed	*5*	*6*
Missing Abs	14	26
- bad density	*7*	*9*
Others	2	2

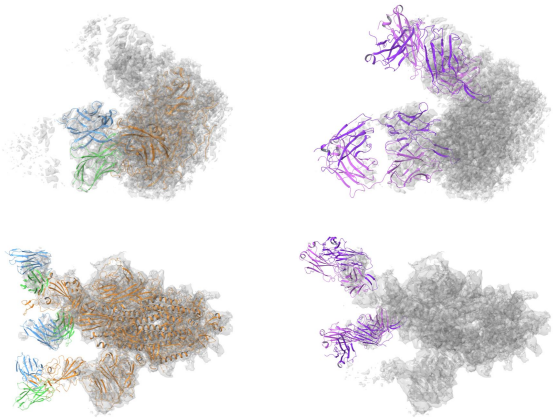


Figure 4: *Left:* An exhaustive classification of our errors on the `sorted` split for both Fabs and VHHs. *Right:* Illustration of the two main modes of failures on two examples. The first row corresponds to `8E8R` and the second to `8HEC`.
Most errors result from predicting more abs - including scenarios where these extra predictions are reasonable like for `8E8R` - or fewer abs - including scenarios where the missing ab falls in a region with very poor density like in `8HEC`.

The vast majority of failures amount to either predicting too many antibodies or not predicting enough.

When predicting too many Abs, interestingly, the right ones are *always* included in our prediction. Moreover, in 5 out of these 15 errors, the extra predictions were found to correspond to an antibody that *actually appears in the density*, such as an element from another asymmetrical unit, and thus can be considered as accurate detections. In the eight remaining cases, two extra predictions result from a wrong ordering (the wrong prediction is ranked first) and four result from a faulty thresholding: the artifacts have lower probabilities but fail to be automatically discarded. Overall, we remark that these additional predictions are both infrequent and can easily be discarded by practitioners.

Furthermore, predictions for 14 systems lack one or more Ab. In over a third of the cases, this could be explained by a poor density of the map around the missing antibody. Forcing `CrAI` to produce additional predictions on those 12 systems, we observe that 16 out of 20 missing Abs can be found, often obtained after just one or two extra predictions. Thus, our software offers the option to predict a certain *number* of predictions instead of relying on automatic thresholding, to allow practitioners to capture potential missing annotations.

In summary, we noticed that most of our errors originate from the ordering and thresholding of the predictions and not from the detection of antibodies.

To further validate this observation, we performed a study of the accuracy of `CrAI` and `dock_in_map` across a varying number of predictions. For this, we forced both methods to output up to ten predictions and then computed the hit rates obtained when keeping the k-best predictions. While our method can easily output ten predictions, to get this result for `dock_in_map` we independently ran it

on a growing set of inputs. The first few inputs correspond to the observed structures of antibodies while additional ones are copies of these structures in a random order.

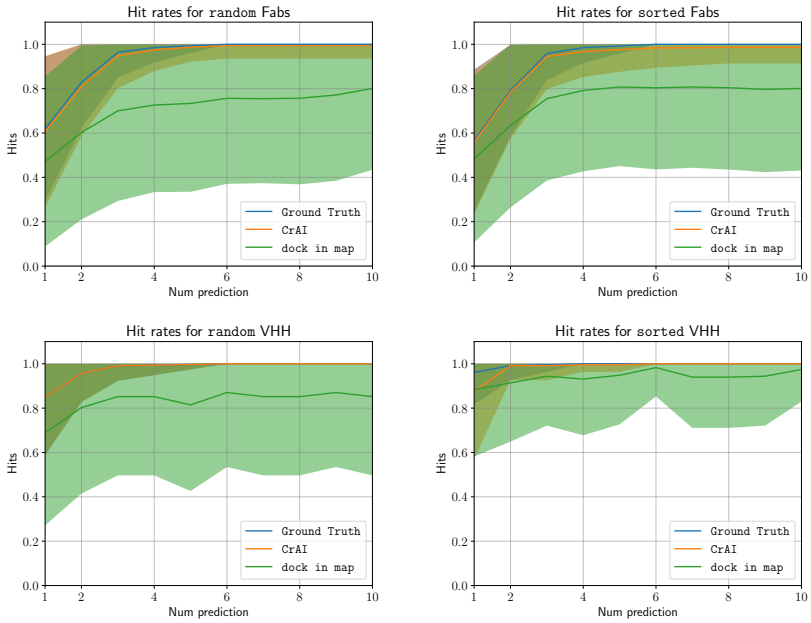


Figure 5: Hits of different approaches (y axis) as a function of the number of predictions (x axis) in our four data settings. The solid lines represent the mean performance of systems and shaded regions correspond to the variance. Even if we observe a small discrepancy between our prediction and the ground truth line, note that most errors originate from the automatic thresholding feature, and our method achieves near perfect hit rate with as few as 6 detections, compared to `dock_in_map`, which fails to capture a significant number of present structures.

In Figure 5 we plot the fraction of all antibodies captured by different approaches, when making a growing number of predictions per system. The blue curve represents the best achievable results. It does not always equal one as some systems contain multiple antibodies. As can be seen in this figure, a small discrepancy exists between the ground truth and our tool. However this discrepancy disappears around $k = 6$ predictions, suggesting that with this number of detections per system our method captures *all* antibodies. This further justifies that most of our errors originate from the thresholding. Thus, if our tool fails to detect an antibody, practitioners can ask for more predictions with a high chance of seeing it predicted. `dock_in_map` has a much wider gap that tends to stay consistent despite allowing it to output more predictions.

Discussion

In this paper, we addressed the problem of automatically finding the Abs in cryo-EM densities. This step currently constitutes a tedious, manual step thus hindering efficient and scalable structure estimation.

To achieve this goal, we proposed a customized solution, which exploits the structural properties and addresses the specific challenges of the problem, such as handling significant data scarcity and heterogeneity. Specifically, we leveraged the conserved structure of Fabs to cast this problem as an object detection one. We gathered and curated a database to enable a data-driven solution. Finally, we then designed a customized pose representation and loss based on optimal transport, which all help integrate prior information, while remaining efficient and flexible. Using our approach, no extra input is required to predict the number, position and pose of Fabs and VHHs in a density.

We validate our results on experimental maps and find the Ab positions with a hit rate above 90%, which represents a 25% (resp 15%) improvement on Fabs (resp VHHs) over existing methods, while requiring no extra inputs and being *thousands of times faster* even when run on a single CPU. We

show that the predicted pose correlates well with experiments, and illustrate our tool's performance on six systems relevant to drug design.

We believe that this tool can reduce the burden on structural biologists working with cryo-EM densities of Abs and accelerate the resolution of 3D complexes of Abs bound to their antigen. In line with this objective, the method ships as a ChimeraX[43] bundle to enable seamless integration.

In the future, it will be interesting to expand our approach to other conserved families, beyond Fabs and VHHs, and enable placing several folded domains into a density, which has so far been done without machine learning, [8, 37]. More broadly, we could replace a fixed template by a parametrized family of embeddings of the folded domains, expanding the expressive power of our framework.

The exceptional throughput of our tool opens the door to finding Abs in the output of heterogeneous reconstruction or even continuous distribution of maps, and thus to capture several modes of antibody binding. Moreover, the automatization of cryo-EM structure resolution by our technique enables the generation of antibody-antigen complexes at a larger scale, a critical step to improve our understanding and detailed modeling of antibody binding. More structural information will open possibilities for better and more accurate antibody modeling, that will speed up drug discovery process of biotherapeutics.

References

[1] M. Arbabi Ghahroudi, A. Desmyter, L. Wyns, R. Hamers, and S. Muyldermans. Selection and identification of single domain antibody fragments from camel heavy-chain antibodies. *FEBS letters*, 414(3):521–526, 1997.

[2] T. Bepler, A. Morin, M. Rapp, J. Brasch, L. Shapiro, A. J. Noble, and B. Berger. TOPAZ: A positive-unlabeled convolutional neural network CryoEM particle picker that can pick any size and shape particle. *Microscopy and Microanalysis*, 25(S2):986–987, aug 2019. doi: 10.1017/s143192761900566x. URL `https://doi.org/10.1017%2Fs143192761900566x`.

[3] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[4] T. L. Blundell and S. Patel. High-throughput x-ray crystallography for drug discovery. *Current Opinion in Pharmacology*, 4(5):490–496, oct 2004. doi: 10.1016/j.coph.2004.04.007. URL `https://doi.org/10.1016%2Fj.coph.2004.04.007`.

[5] J. Bouvette, Q. Huang, A. A. Riccio, W. C. Copeland, A. Bartesaghi, and M. J. Borgnia. Automated systematic evaluation of cryo-EM specimens with SmartScope. *eLife*, 11, aug 2022. doi: 10.7554/elife.80047. URL `https://doi.org/10.7554%2Felife.80047`.

[6] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[7] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas. Persistence barcodes for shapes. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 124–135, 2004.

[8] L. Chang, F. Wang, K. Connolly, H. Meng, Z. Su, V. Cvirkaite-Krupovic, M. Krupovic, E. H. Egelman, and D. Si. Deeptracer-id: De novo protein identification from cryo-em maps. *Biophysical Journal*, 121(15):2840–2848, 2022.

[9] F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Persistence-based clustering in riemannian manifolds. *Journal of the ACM (JACM)*, 60(6):1–38, 2013.

[10] Y. Chen, Y.-N. Zhang, R. Yan, G. Wang, Y. Zhang, Z.-R. Zhang, Y. Li, J. Ou, W. Chu, Z. Liang, Y. Wang, Y.-L. Chen, G. Chen, Q. Wang, Q. Zhou, B. Zhang, and C. Wang. Ace2-targeting monoclonal antibody as potent and broad-spectrum coronavirus blocker. *Signal Transduction and Targeted Therapy*, 6(1), Aug. 2021. ISSN 2059-3635. doi: 10.1038/s41392-021-00740-y. URL `http://dx.doi.org/10.1038/s41392-021-00740-y`.

[11] Z. Chen, S. Watanabe, H. Hashida, M. Inoue, Y. Daigaku, M. Kikkawa, and K. Inaba. Cryo-em structures of human spca1a reveal the mechanism of ca 2+ /mn 2+ transport into the golgi apparatus. *Science Advances*, 9(9), Mar. 2023. ISSN 2375-2548. doi: 10.1126/sciadv.add9742. URL `http://dx.doi.org/10.1126/sciadv.add9742`.

[12] M. L. Chiu, D. R. Goulet, A. Teplyakov, and G. L. Gilliland. Antibody structure and function: the basis for engineering therapeutics. *Antibodies*, 8(4):55, 2019.

[13] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 424–432. Springer, 2016.

[14] T. Cohen, M. Halfon, and D. Schneidman-Duhovny. Nanonet: Rapid and accurate end-to-end nanobody modeling by deep learning. *Frontiers in immunology*, 13:958584, 2022.

[15] V. I. Cushing, A. F. Koh, J. Feng, K. Jurgaityte, A. K. Bahl, S. Ali, A. Kotecha, and B. J. Greber. High-resolution cryo-electron microscopy of the human cdk-activating kinase for structure-based drug design. *bioRxiv*, pages 2023–04, 2023.

[16] W. L. DeLano. The pymol molecular graphics system. *http://www. pymol. org/*, 2002.

[17] S. Duggan. Caplacizumab: first global approval. *Drugs*, 78:1639–1642, 2018.

[18] J. Dunbar, K. Krawczyk, J. Leem, T. Baker, A. Fuchs, G. Georges, J. Shi, and C. M. Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42(D1):D1140–D1146, 2014.

[19] C. Emmons and L. Hunsicker. Muromonab-cd3 (orthoclone okt3): the first monoclonal antibody approved for therapeutic use. *Iowa Medicine: Journal of the Iowa Medical Society*, 77(2): 78–82, 1987.

[20] P. Emsley, B. Lohkamp, W. G. Scott, and K. Cowtan. Features and development of coot. *Acta Crystallographica Section D: Biological Crystallography*, 66(4):486–501, 2010.

[21] Q. Fan, Y. Li, Y. Yao, J. Cohn, S. Liu, S. M. Vos, and M. A. Cianfrocco. Cryorl: Reinforcement learning enables efficient cryo-em data collection. *arXiv preprint arXiv:2204.07543*, 2022.

[22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[23] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019.

[24] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[25] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[26] M. He, X. Chi, Z. Zha, Y. Li, J. Chen, Y. Huang, S. Huang, M. Yu, Z. Wang, S. Song, X. Liu, S. Wei, Z. Li, T. Li, Y. Wang, H. Yu, Q. Zhao, J. Zhang, Q. Zheng, Y. Gu, S. Li, and N. Xia. Structural basis for the shared neutralization mechanism of three classes of human papillomavirus type 58 antibodies with disparate modes of binding. *Journal of Virology*, 95 (7), Mar. 2021. ISSN 1098-5514. doi: 10.1128/jvi.01587-20. URL `http://dx.doi.org/10.1128/jvi.01587-20`.

[27] A. M. Hummer, B. Abanades, and C. M. Deane. Advances in computational structure-based antibody design. *Current Opinion in Structural Biology*, 74:102379, 2022.

[28] K. Jamali, L. Käll, R. Zhang, A. Brown, D. Kimanius, and S. H. Scheres. Automated model building and protein identification in cryo-em maps. *Nature*, pages 1–2, 2024.

[29] P. T. Jones, P. H. Dear, J. Foote, M. S. Neuberger, and G. Winter. Replacing the complementarity-determining regions in a human antibody with those from a mouse. *Nature*, 321(6069):522–525, 1986.

[30] I. Jovčevska and S. Muyldermans. The therapeutic potential of nanobodies. *BioDrugs*, 34(1): 11–26, 2020.

[31] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[32] S. Kaji, T. Sudo, and K. Ahara. Cubical ripser: Software for computing persistent homology of image and volume data. *arXiv preprint arXiv:2005.12692*, 2020.

[33] H. Kaplon, S. Crescioli, A. Chenoweth, J. Visweswaraiah, and J. M. Reichert. Antibodies to watch in 2023. In *MAbs*, volume 15, page 2153410. Taylor & Francis, 2023.

[34] D. Kimanius, L. Dong, G. Sharov, T. Nakane, and S. H. W. Scheres. New tools for automated cryo-EM single-particle analysis in RELION-4.0. *Biochemical Journal*, 478(24):4169–4185, dec 2021. doi: 10.1042/bcj20210708. URL `https://doi.org/10.1042%2Fbcj20210708`.

[35] G. Köhler and C. Milstein. Continuous cultures of fused cells secreting antibody of predefined specificity. *Nature*, 256(5517):495–497, 1975.

[36] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[37] D. Liebschner, P. V. Afonine, M. L. Baker, G. Bunkóczi, V. B. Chen, T. I. Croll, B. Hintze, L.-W. Hung, S. Jain, A. J. McCoy, et al. Macromolecular structure determination using x-rays, neutrons and electrons: recent developments in phenix. *Acta Crystallographica Section D: Structural Biology*, 75(10):861–877, 2019.

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[39] J. Maharana, P. Sarma, M. K. Yadav, S. Saha, V. Singh, S. Saha, M. Chami, R. Banerjee, and A. K. Shukla. Structural snapshots uncover a key phosphorylation motif in gpcrs driving β-arrestin activation. *Molecular Cell*, 83(12):2091–2107.e7, June 2023. ISSN 1097-2765. doi: 10.1016/j.molcel.2023.04.025. URL `http://dx.doi.org/10.1016/j.molcel.2023.04.025`.

[40] S. Matsumoto, S. Ishida, M. Araki, T. Kato, K. Terayama, and Y. Okuno. Extraction of protein dynamics information from cryo-EM maps using deep learning. *Nature Machine Intelligence*, 3(2):153–160, feb 2021. doi: 10.1038/s42256-020-00290-y. URL `https://doi.org/10.1038%2Fs42256-020-00290-y`.

[41] J. McCafferty, A. D. Griffiths, G. Winter, and D. J. Chiswell. Phage antibodies: filamentous phage displaying antibody variable domains. *Nature*, 348(6301):552–554, 1990.

[42] M. Peplow. Cryo-electron microscopy makes waves in pharma labs. *Nature Reviews Drug Discovery*, 16(12):815–818, 2017.

[43] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. UCSF chimera?a visualization system for exploratory research and analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, 2004. doi: 10.1002/jcc.20084. URL `https://doi.org/10.1002%2Fjcc.20084`.

[44] E. F. Pettersen, T. D. Goddard, C. C. Huang, E. C. Meng, G. S. Couch, T. I. Croll, J. H. Morris, and T. E. Ferrin. Ucsf chimerax: Structure visualization for researchers, educators, and developers. *Protein Science*, 30(1):70–82, 2021.

[45] G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[46] J. Pfab, N. M. Phan, and D. Si. Deeptracer for fast de novo cryo-em protein structure modeling and special studies on cov-related complexes. *Proceedings of the National Academy of Sciences*, 118(2):e2017525118, 2021.

[47] M. Pirazzini, A. Grinzato, D. Corti, S. Barbieri, O. Leka, F. Vallese, M. Tonellato, C. Silacci-Fregni, L. Piccoli, E. Kandiah, G. Schiavo, G. Zanotti, A. Lanzavecchia, and C. Montecucco. Exceptionally potent human monoclonal antibodies are effective for prophylaxis and treatment of tetanus in mice. *Journal of Clinical Investigation*, 131(22), Nov. 2021. ISSN 1558-8238. doi: 10.1172/jci151676. URL `http://dx.doi.org/10.1172/JCI151676`.

[48] A. Punjani, J. L. Rubinstein, D. J. Fleet, and M. A. Brubaker. cryosparc: algorithms for rapid unsupervised cryo-em structure determination. *Nature methods*, 14(3):290–296, 2017.

[49] C. R. Qi, O. Litany, K. He, and L. J. Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019.

[50] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[51] R. Sanchez-Garcia, J. Segura, D. Maluenda, C. Sorzano, and J. Carazo. MicrographCleaner: A python package for cryo-EM micrograph cleaning using deep learning. *Journal of Structural Biology*, 210(3):107498, jun 2020. doi: 10.1016/j.jsb.2020.107498. URL `https://doi.org/10.1016%2Fj.jsb.2020.107498`.

[52] S. H. Scheres. Relion: implementation of a bayesian approach to cryo-em structure determination. *Journal of structural biology*, 180(3):519–530, 2012.

[53] C. Schneider, M. I. Raybould, and C. M. Deane. Sabdab in the age of biotherapeutics: updates including sabdab-nano, the nanobody structure tracker. *Nucleic acids research*, 50(D1):D1368–D1372, 2022.

[54] G. P. Smith. Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface. *Science*, 228(4705):1315–1317, 1985.

[55] TheAntibodySociety. The antibody society. therapeutic monoclonal antibodies approved or in review in the eu or us. 8/23; www.antibodysociety.org/resources/approved-antibodies., 2023.

[56] L. G. Trabuco, E. Villa, E. Schreiner, C. B. Harrison, and K. Schulten. Molecular dynamics flexible fitting: a practical guide to combine cryo-electron microscopy and x-ray crystallography. *Methods*, 49(2):174–180, 2009.

[57] C. Tralie, N. Saul, and R. Bar-On. Ripser.py: A lean persistent homology library for python. *The Journal of Open Source Software*, 3(29):925, Sep 2018. doi: 10.21105/joss.00925. URL `https://doi.org/10.21105/joss.00925`.

[58] T. Wagner, F. Merino, M. Stabrin, T. Moriya, C. Antoni, A. Apelbaum, P. Hagel, O. Sitsel, T. Raisch, D. Prumbaum, D. Quentin, D. Roderer, S. Tacke, B. Siebolds, E. Schubert, T. R. Shaikh, P. Lill, C. Gatsogiannis, and S. Raunser. SPHIRE-crYOLO is a fast and accurate fully automated particle picker for cryo-EM. *Communications Biology*, 2(1), jun 2019. doi: 10.1038/s42003-019-0437-z. URL `https://doi.org/10.1038%2Fs42003-019-0437-z`.

[59] F. Wang, H. Gong, G. Liu, M. Li, C. Yan, T. Xia, X. Li, and J. Zeng. DeepPicker: A deep learning approach for fully automated particle picking in cryo-EM. *Journal of Structural Biology*, 195(3):325–336, sep 2016. doi: 10.1016/j.jsb.2016.07.006. URL `https://doi.org/10.1016%2Fj.jsb.2016.07.006`.

[60] R. Yao, J. Qian, and Q. Huang. Deep-learning with synthetic data enables automated picking of cryo-EM particle images of biological macromolecules. *Bioinformatics*, 36(4):1252–1259, oct 2019. doi: 10.1093/bioinformatics/btz728. URL `https://doi.org/10.1093%2Fbioinformatics%2Fbtz728`.

[61] J. Zhang, Z. Wang, Y. Chen, R. Han, Z. Liu, F. Sun, and F. Zhang. PIXER: an automated particle-selection method based on segmentation using a deep neural network. *BMC Bioinformatics*, 20(1), jan 2019. doi: 10.1186/s12859-019-2614-y. URL https://doi.org/10.1186%2Fs12859-019-2614-y.

[62] E. D. Zhong, T. Bepler, B. Berger, and J. H. Davis. CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks. *Nature Methods*, 18(2):176–185, feb 2021. doi: 10.1038/s41592-020-01049-4. URL https://doi.org/10.1038%2Fs41592-020-01049-4.

[63] Y. Zhou, H. Daver, B. Trapkov, L. Wu, M. Wu, K. Harpsøe, P. R. Gentry, K. Liu, M. Larionova, J. Liu, N. Chen, H. Bräuner-Osborne, D. E. Gloriam, T. Hua, and Z.-J. Liu. Molecular insights into ligand recognition and g protein coupling of the neuromodulatory orphan receptor gpr139. *Cell Research*, 32(2):210–213, Dec. 2021. ISSN 1748-7838. doi: 10.1038/s41422-021-00591-w. URL http://dx.doi.org/10.1038/s41422-021-00591-w.

[64] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023.

Methods

Building a database

We build a curated antibody database to train and test our method, comprising Fabs and VHHs. Fabs are composed of one constant and one variable domain for each of the heavy and the light chains. The two variable domains are denoted as the Fv variable fragment (Fv). VHHs are antibody fragments that correspond to the sole variable domain of the heavy chain. They represent a promising family for therapeutic antibodies.

The Fab data is originally fetched from *SabDab*[18] in the form of a list of protein chains. We fix a few broken annotations, notably for systems containing both Fabs and VHHs (more details in Section A.3). Using the PDB[3], we find the corresponding cryo-EM maps and download all corresponding maps and structures. The maps contain densities of both Fabs and Fvs, but the constant region of Fabs is often missing in the deposited structure. To avoid false negatives in our data, we chose to predict the position of the Fv as this is the only part consistently reported. Then, we remove systems with resolution below 10Å or ones with no antibodies or antigen chains, and group all hits pertaining to the same structure together, yielding a total of 1032 systems.

The resulting map files can be enormous (up to 10^9 grid cells) especially for symmetric assemblies, such as viruses. Since the asymmetric unit only occupies a fraction of the map, using whole maps would create false negatives. Indeed, some regions of the map correspond to proteins omitted in the deposited structure (e.g. pdb `7kcr`). Hence, we crop the original maps around the structure with a margin of 25Å. Additionally, we resample maps to a fixed voxel size of two Å and normalize them by zeroing out negative values and dividing by the maximum of the map.

We split this dataset following a temporal splitting strategy with more recent systems in the test split (`sorted`). While this procedure is often used to provide a realistic use case, it can also introduce a bias, for instance towards COVID structures. Hence, we also report our performance following a random split (`random`). We obtain 722, 155 and 155 systems in the train, validation and test splits respectively. Note that a single system can include several Fabs. On average, there are 2.25 Fabs per system in our dataset. The number of Fabs per split is 1,627, 320 and 374 respectively - with similar numbers for the `random` split. We now have a split database of cropped, resampled, normalized cryo-EM density maps containing Fabs.

All of these steps are repeated to create a VHH database. We started from *SAbDab-nano*[53] and applied the same filters on the raw data, resulting in 398 systems. Then, we performed the *random* and *sorted* splitting procedures as above. This amounts to 278, 60, 60 `systems` in each split and 458, 74, 86 when counted as `ab`, for a mean of 1.55 VHHs per system - again with similar numbers for the `random` split.

Motivation

The design of our approach, `CrAI`, is motivated by several methodological challenges. First, due to the challenges inherent in the limited scale and significant noise present in the training data, we incorporate the conserved structure of antibodies as prior information to make learning more data-efficient and robust. Second, to facilitate antibody pose finding in the presence of limited data, we use a custom parametrization of the rotation that takes into account an *expected* pose, while being flexible to allow arbitrary orientations. Third, we introduce a fully convolutional design to accommodate arbitrary grid sizes that might be present at inference time. We also train the network with rotation augmentation to approximate rotation equivariance and accommodate arbitrary orientations. Finally, we introduce a custom loss when training our model. Our loss incorporates a formulation based on Optimal Transport along with persistence diagrams to better capture the geometric aspects of our problem, such as predicting non-overlapping objects and including *distance*-based penalties between our prediction and the ground truth. Finally, we introduce a Non-Maximum Suppression step based on persistence diagrams, which selects the most likely individual detections given a predicted probability map.

Problem formulation

Starting from an input cryo-EM map, we want our method to output the 3D coordinates of one or several Abs. Because of the highly conserved structure of Abs, we can simplify our problem by only predicting how to align a fixed antibody template $\mathbf{T}$ with the Abs, without deformations. We used the Fab structure in the pdb `7lo8` as a template and computed the optimal alignments with `pymol align`[16] during data pre-processing. We provide an example alignment of our template in Figure 1A.

More formally: Let $\mathcal{X}$ be the cryo-EM density we consider, $n_{\mathcal{X}}$ the number of Abs contained in this density and $\mathcal{A}_{\mathcal{X}} = \{\mathbf{A}_i, 0 \leq i < n_{\mathcal{X}}\}$ the set of such Abs. Given an Ab $\mathbf{A}$ and a registration objective d, let $R^*_{\mathbf{A}}, T^*_{\mathbf{A}} = \min_{R,T \in (SO_3(\mathbb{R}) \times \mathbb{R}^3)} d(\mathbf{A}, R\mathbf{T} + T)$ be the translation and rotation that best align $\mathbf{T}$ to $\mathbf{A}$. Finally let $\mathcal{S}_{\mathcal{X}} = \{(R^*_{\mathbf{A}_i}, T^*_{\mathbf{A}_i}), \mathbf{A}_i \in \mathcal{A}_{\mathcal{X}}\}$ be the set of optimal alignments. Note $\mathcal{S}_{\mathcal{X}}$ consists of elements of the Euclidean group in 3D that is six-dimensional, whereas elements of $\mathcal{A}_{\mathcal{X}}$ are three-dimensional coordinates for hundreds of atoms. In this paper, we aim to predict $\mathcal{S}_{\mathcal{X}}$ instead of $\mathcal{A}_{\mathcal{X}}$.

The optimal rotations mentioned above can be parametrized in many ways. Let $\vec{p}_{\mathbf{A}}$ denote the unit vector oriented from the center of mass of an Ab $\mathbf{A}$ towards its antigen. Since canonical binding tends to happen through the CDRs, we observe that it is easier to predict $\vec{p}_{\mathbf{A}}$ than the rotation around $\vec{p}_{\mathbf{A}}$ from the density. Therefore, we decomposed $R^*_{\mathbf{A}}$ into a rotation transforming $\vec{p}_{\mathbf{T}}$ into $\vec{p}_{\mathbf{A}}$ and one 2D rotation around $\vec{p}_{\mathbf{T}}$ of angle $\theta_{\mathcal{A}}$. The generality of this decomposition is established in Supplemental A.1 and its relevance is shown in Figure 2.

Our problem is now formulated as an object detection problem, which consists of detecting, localizing and aligning the Abs in a given density. Following common practice in the object detection literature [50], we overlay an occupancy grid $\mathcal{G}^{\mathcal{X},\mathbf{S}}$ of size $\mathbf{S}$ over our input: this grid contains ones for cells encompassing an Ab and zeros elsewhere. The size of this grid cells corresponds to a fixed spatial location and not on the resolution of the map. We then decompose each translation into two parts: one going to the corner of the occupied cell and one from this corner to the Ab: $T^*_{\mathbf{A}} = T^{cell}_{\mathbf{A}} + T^{loc}_{\mathbf{A}}$, as shown in Figure 1A. Hence, finding the optimal translation comprises finding occupied cells and local translations in those cells. This decomposition makes the encoding of the output invariant to translation and avoids manipulating large values to encode translations in the grid.

Architecture and learning procedure We aim to solve the object detection problem stated above with a Machine Learning approach, trained on our dataset. Given a cryo EM density of size S_i, $\mathcal{X} \in \mathbb{R}^{S_i}$ and its corresponding occupancy grid of size $\mathbf{S}$, $\mathcal{G}^{\mathcal{X},\mathbf{S}}$, our network is a function $f_\theta : \mathbb{R}^{S_i} \to \mathbb{R}^{10 \times \mathbf{S}}$ such that $\hat{y} = f_\theta(\mathcal{X}) \in \mathbb{R}^{10 \times \mathbf{S}}$ is our prediction for $\mathcal{X}$. The prediction at a position s is denoted as $\hat{y}^k(s) \in \mathbb{R}^{10}$, where $k = 1..10$. The first dimension of this output, $\hat{y}^0(.)$, is a prediction of the occupancy grid $\mathcal{G}^{\mathcal{X},\mathbf{S}}$. The nine other dimensions are predictions in each cell relative to the putative Ab contained in it. The details about both the exact role of each dimension as well as our training procedure are described below.

The architecture of our model f_θ is a 3D UNet [13] with a depth of 4. To enhance robustness, the network is trained using data augmentation, with the eight possible rotations over a grid and random cropping inside the input grid up to three cells. We used Adam optimizer over 1000 epochs. Hyper-parameters and exact architecture were not extensively tuned to avoid artificially boosting performance. Finer details about the ones we used can be found in Supplementary A.2.

Prediction of the right cells using optimal transport For convenience, in this section, we drop indices and denote the ground truth occupancy grid $\mathcal{G}^{\mathcal{X},\mathbf{S}}$ as $\mathcal{G}$. The first slice of our output $\hat{y}$ is a prediction of the occupancy grid and hence, let us denote it as $\hat{\mathcal{G}}_s = \hat{y}^0(s)$. In order to make $\hat{\mathcal{G}}$ close to $\mathcal{G}$, we will use two loss terms.

Because most grid cells are unoccupied, our prediction is very imbalanced. Hence, our first loss term is a weighted binary focal loss [38] that focuses on cells with *wrong* predictions :

$$\mathcal{L}_1(\hat{\mathcal{G}}, \mathcal{G}) = \sum_{s \leq \mathbf{S}} (\lambda * \mathcal{G}_s * (1 - \hat{\mathcal{G}}_s)^\gamma * \log(\hat{\mathcal{G}}_s) + (1 - \mathcal{G}_s) * \hat{\mathcal{G}}_s^\gamma * \log(1 - \hat{\mathcal{G}}_s)). \tag{1}$$

We used $\lambda = 30$ and $\gamma = 4$ in our experiments.

We observe that this focal loss $\mathcal{L}_1$ does not consider the *distance* between our prediction and the ground truth. Predicting the neighbor pixel results in the same loss value as predicting the opposite side of the map. To address this issue, we add an *optimal transport* term based on Sinkhorn divergence to our loss [45]. After normalization, we can view $\hat{\mathcal{G}}$ and $\mathcal{G}$ as measures defined over the regular grid $\mathbf{g}$. The optimal transport cost between those measures is defined as :

$$\mathrm{OT}(\hat{\mathcal{G}}, \mathcal{G}) \overset{\text{def.}}{=} \min_{\pi \in \Pi_{\hat{\mathcal{G}},\mathcal{G}}} \int_{\mathbf{g}^2} \|x-y\|^2 d\pi(x,y).$$

Where $\Pi_{\hat{\mathcal{G}},\mathcal{G}}$ is the set of coupling measures on $\mathbf{g}$ with marginals $\hat{\mathcal{G}}$ and $\mathcal{G}$. The actual computation of this term is prohibitively expensive, hence we add a regularization term with weight $\varepsilon > 0$ and introduce :

$$\begin{aligned}\mathrm{OT}_\varepsilon(\hat{\mathcal{G}}, \mathcal{G}) &\overset{\text{def.}}{=} \min_{\pi \in \Pi_{\hat{\mathcal{G}},\mathcal{G}}} \int_{\mathcal{X}^2} \|x-y\|^2 d\pi(x,y) + \varepsilon \mathrm{KL}(\pi \mid \hat{\mathcal{G}} \otimes \mathcal{G}) \\ &\text{where } \mathrm{KL}(\pi \mid \hat{\mathcal{G}} \otimes \mathcal{G}) \overset{\text{def.}}{=} \int_{\mathcal{X}^2} \log\left(\frac{\mathrm{d}\pi}{\mathrm{d}\hat{\mathcal{G}}\mathrm{d}\mathcal{G}}\right) \mathrm{d}\pi.\end{aligned}$$

We notice that $\mathrm{OT}_0 = \mathrm{OT}$ and for any $\varepsilon > 0$, adding this $\varepsilon\mathrm{KL}$ regularization makes the computation tractable. However, this regularization term also introduces the so-called *entropic bias* since for a measure x, $\mathrm{OT}_\varepsilon(x,x) > 0$. Sinkhorn divergences were introduced to correct for this bias while remaining tractable, and that is what we use as a loss :

$$\mathcal{L}_2(\hat{\mathcal{G}}, \mathcal{G}) = \mathrm{OT}_\varepsilon(\hat{\mathcal{G}}, \mathcal{G}) - \frac{1}{2}\mathrm{OT}_\varepsilon(\hat{\mathcal{G}}, \hat{\mathcal{G}}) - \frac{1}{2}\mathrm{OT}_\varepsilon(\mathcal{G}, \mathcal{G})$$

This term gives meaningful supervision to all voxels of our grids and depends on the distance to the closest occupied voxels. We used the *GeomLoss* package to compute it and refer to its corresponding paper [23] for a detailed discussion of the computation. The relevance of this loss is assessed in the results section.

Prediction within each cell Let $\mathbf{A}$ be an antibody in our system and $s_\mathbf{A}$ its position in the grid. Beyond predicting the right grid cell, we also want finer grained prediction about its precise position in the cell $T^{loc}_\mathbf{A}$, its orientation $(\vec{p}_\mathbf{A}, \theta_\mathbf{A})$ and its classification as a Fab or a VHH. Let $\hat{y}^j_\mathbf{A} = \hat{y}^j(s_\mathbf{A}) \in \mathbb{R}^{10}$ be the prediction at this position, we will introduce additional loss terms to capture these finer grained predictions. We emphasize that these will only be applied on grid cells containing an antibody. We use the notation $\hat{y}^{j:j+k-1}_\mathbf{A}$ to denote the k dimensional vector obtained from the concatenation of $\hat{y}^j_\mathbf{A}, \hat{y}^{j+1}_\mathbf{A} \cdots \hat{y}^{j+k-1}_\mathbf{A}$.

To predict the right pose of $\mathbf{A}$ in the cell, we use eight values. The first three are used to predict the offset from the corner of the grid cell, $T^{loc}_\mathbf{A}$. We use a mean squared error loss to learn these values, $\mathcal{L}_3$. The following three values are used to predict $\vec{p}_\mathbf{A}$ by directly predicting its coordinates. The corresponding loss, $\mathcal{L}_4$, is composed of a dot product term to control the direction of the prediction along with a term to make this vector unit norm. Favoring unit norms avoids numerically unstable normalizations. Finally, the remaining two values are used to predict the angle $\theta_\mathbf{A}$. Instead of directly predicting the angle, we aim to predict $u_\mathbf{A} \in \mathbb{R}^2$, the vector of polar coordinates $(1, \theta_\mathbf{A})$. This formulation avoids singularities and was shown to be beneficial when predicting angles [31]. Hence, $\mathcal{L}_5$ has a similar form than $\mathcal{L}_4$ in two dimensions to predict $u_\mathbf{A}$. We end up with the following losses:

$$\mathcal{L}_3(\hat{y}^{1:3}_\mathbf{A}, T^{loc}_\mathbf{A}) = \mathtt{mse}(\hat{y}^{1:3}_\mathbf{A}, T^{loc}_\mathbf{A}), \tag{2}$$

$$\mathcal{L}_4(\hat{y}^{4:6}_\mathbf{A}, p_\mathbf{A}) = 1 - \langle \hat{y}^{4:6}_\mathbf{A}, p_\mathbf{A} \rangle + \mathtt{mse}(\|\hat{y}^{4:6}_\mathbf{A}\|, 1), \tag{3}$$

$$\mathcal{L}_5(\hat{y}^{7:8}_\mathbf{A}, u_\mathbf{A}) = 1 - \langle \hat{y}^{7:8}_\mathbf{A}, u_\mathbf{A} \rangle + \mathtt{mse}(\|\hat{y}^{7:8}_\mathbf{A}\|, 1). \tag{4}$$

Finally, as we use a single model for both Fabs and VHHs, we have a term that represents the probability that the object contained in the grid cell is a VHH and not a Fab. Let $\delta_n(x)$ be the

indicator function for VHHs (one if VHH else zero). We again construct a weighted focal loss, with a weight of $\lambda_n = 1000/400$ corresponding to the ratio of VHHs to Fabs, yielding the last loss,

$$\mathcal{L}_6(\hat{y}^9_{\mathbf{A}}, \mathbf{A}) = \lambda_n * \delta_n(\mathbf{A}) * (1 - \hat{y}^9_{\mathbf{A}})^\gamma * \log(\hat{y}^9_{\mathbf{A}}) + (1 - \delta_n(\mathbf{A})) * (\hat{y}^9_{\mathbf{A}})^\gamma * \log(1 - \hat{y}^9_{\mathbf{A}})). \quad (5)$$

To train our network we use a weighted sum ($\lambda_s = 0.2$) of previous loss terms as the final loss. We sum the first two global ones and the sum of the four others over each antibody in our system:

$$\mathcal{L}_{tot}(\hat{y}, \mathcal{X}) = \sum_{i=1}^{2} \mathcal{L}_i(\hat{\mathcal{G}}, \mathcal{G}) + \lambda_s * \sum_{i=3}^{6} \sum_{\mathbf{A} \in \mathcal{A}_\mathcal{X}} \mathcal{L}_i(\hat{y_{\mathbf{A}}}, \mathbf{A}). \quad (6)$$

Post processing A well-known problem with object detection is the possibility that the network predicts overlapping objects. Supposing occupancy grid cells edges were 3Å long, adjacent cells could not both contain a Fab. Hence, high values for adjacent cells typically amount to the detection of the same underlying object. Non Maximal Suppression (NMS) algorithms are used to discard such redundant predictions. Starting with our grid $\hat{y}_0$, we want to obtain a list of the distinct local minima. In this paper, we used an approach based on `persistence diagrams` [9], implemented with *cripser*[32, 57]. Simply put, we decrease a threshold probability value from the maximum value of our grid $\hat{y}_0$ and keep track of cells above this threshold. When the value of a cell goes over the threshold, either it has no neighbors belonging to a visited connected component, giving *birth* to a new one, or all neighboring components are merged into the one with lowest initial values and merged ones *die*. The difference between the values of death and birth are called lifetimes. We return connected components sorted by lifetimes.

This procedure takes into account both the *value* of a minimum and its location with respect to other minima. If we suppose that the number of objects to find is known, we keep proceeding until this number is reached (and refer to this setting as and `num`). Otherwise we use a lifetime threshold and retain all predictions above this threshold. We use a threshold of 0.2 that was found to work best on the validation set (see Supplementary A.4). We will denote this setting `thresh`. Interestingly, this procedure allows us to automatically detect the numbers of Abs in a density, without any prior information.

From our density map, we now have a list of predicted values. For each of those, we choose a template based on the classification in Fabs and VHHs and move this template to the predicted location and pose. We save the result as a file in PDB format.

Acknowledgments and Disclosure of Funding

This research was supported by Sanofi. V. M. and M. O. are supported by DataIA, ANR AI Chair AIGRETTE and the ERC Starting Grant EXPROTEA. C.R. and H.M. are employees of Sanofi and may hold shares in the company.

Appendix

A Methods

A.1 Proof of the validity of the decomposition

Following our notations, $R^*_{\mathbf{A}}$ is a rotation that aligns $\mathbf{A}$ onto $\mathbf{T}$. We want to show that $R^*_{\mathbf{A}}$ can be parametrized by the transformation of $p_{\mathbf{T}}$ into $p_{\mathbf{A}}$ along with a rotation around $p_{\mathbf{T}}$.

By construction :

$$\mathbf{A} = R^*_{\mathbf{A}}\mathbf{T} \Rightarrow p_{\mathbf{A}} = R^*_{\mathbf{A}} p_{\mathbf{T}}. \tag{7}$$

Let M be a rotation matrix of axis $p_{\mathbf{A}} \times p_{\mathbf{T}}$ and of angle ρ the angle between these vectors, so that it transforms one into the other. We can write :

$$p_{\mathbf{A}} = M p_{\mathbf{T}} \tag{8}$$

$$M^{-1} R^*_{\mathbf{A}} p_{\mathbf{T}} = p_{\mathbf{T}}. \tag{9}$$

Therefore, $M^{-1}R^*_{\mathbf{A}}$ is a rotation around the axis $p_{\mathbf{T}}$ by some angle θ, and we can write :

$$R^*_{\mathbf{A}} = M Rot_{p_{\mathbf{T}},\theta}. \tag{10}$$

A.2 Detailed architecture

We used a 3D Unet architecture. First let us define a `conv_block`(i,o) as three series of convolutions (with kernel size three and stride one), batch normalization and PreLu activations. The first convolution goes from i to o channels and the two others have o as a fixed number of channels.

In the encoding stage, we use a series of four `conv_blocks` of successive dimensions (1, 32, 64, 128, 256), alternated by a max pooling of kernel size 2 and stride 2.

In the decoding stage, we apply a transpose convolution (256, 256) with kernel size of 3 and stride 2 to the output of the encoder. We concatenate the result with the output from the third encoding `conv_block` to get 384 channels. We then run a `conv_block`(384, 128) on this concatenation.

Finally, we apply a last `conv_block`(128, 128) with only two series on the resulting tensor and replace the third one by a linear layer going from dimension 128 to dimension 10. The classification output dimensions are followed by sigmoid activation.

A.3 Modifications to the SabDab database

We removed a few non canonical systems, such as fusion proteins or `8gq5` that contains 16 VHHs arranged in an unusual way and the obsolete `7ny5`. We then fixed edgy situations, often encountered when VHHs and Fabs are present in the system.

- When querying the Fabs, some chains were wrongly considered separate : `7YVN chain HI 7YVP chains IJ, 8HHX chains HI, 8HHY chains FW-GI, 8DXS chains FI-GJ, 7XOD chains RS-UV-XY, 7MLV chains KF, 6XJA (missing chain AB and missing antigen)`
- VHHs were present when querying Fabs. They were annotated as having just one chain and no antigen partner : `7zlg-h-i-j chain K bound to L, 8hii-j-k chain N bound to L, 7xw6 chain N bound to AB, 7sk7 chain K bound to C, 7sk5 chain E bound to C, 7zyi chain K bound to L, 7xod chain T-W-Z bound to S-V-Y, 6ni2 chain A bound to B-V, 7wpe chain W-Z bound to V-Y, 6ww2 chain K bound to L, 7jhh chain N bound to L, 7ul3 chain C bound to A, 7tuy chain K bound to L`
- Fabs were present when querying VHHs (often interacting with this VHH) : `7pij chain HL bound to N, 7wpd chain XY bound to A, 7wpf chain RS, UV, XY, 7wpf chain AC-HL bound to EN, 7php chain HL bound to N, 7m74 chain HL bound to A`

- VHHs with no antigens were present in the VHH file (actually always bound to a Fab) : `7pij K bound to L, 7wpd Z bound to Y, 7nkr F bound to A, 7wpf T-W-Z bound to S-V-Y, 7wpf chain K-F bound to L-C, 7php-7m74-7jhg chain K, N, N bound to L, 7nis-7nk1-7nk2-7nk6-7nkc-7nj4-7nka chain F bound to A/B-A-A/B-A/B-B/C-A-A`

A.4 Finding the optimal threshold

We present the error in the number of Abs detected for different thresholds in Figure S.1. Throughout datasets, persistence diagrams seem to give a better estimate. The overall best value is achieved for a threshold of approximately 0.2 and is not very sensitive to this threshold.

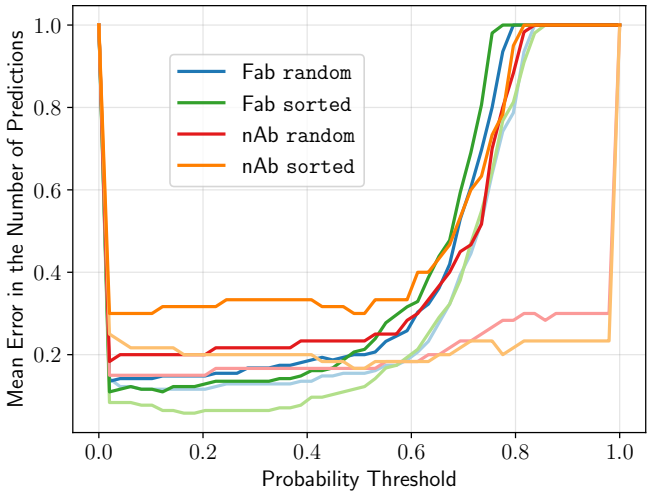


Figure S.1: Finding the best threshold for selecting the right number of system. x-axis represents the threshold value. y-axis represents the mean absolute error in the number of predicted systems over the data set. Different colors represent different data sets. Darker lines represent a naïve approach for NMS while the lighter ones are computed using persistence diagrams.

B Results

B.1 Fab detection performance

We compute the distances from prediction to ground truth obtained from our tool and `dock_in_map`. For a nicer visualization, we cap all distances to 20Å. We also count failed systems and errors as 20Å predictions so that the bar at 20Å represents all failures of a tool. We show the result in Figure S.2

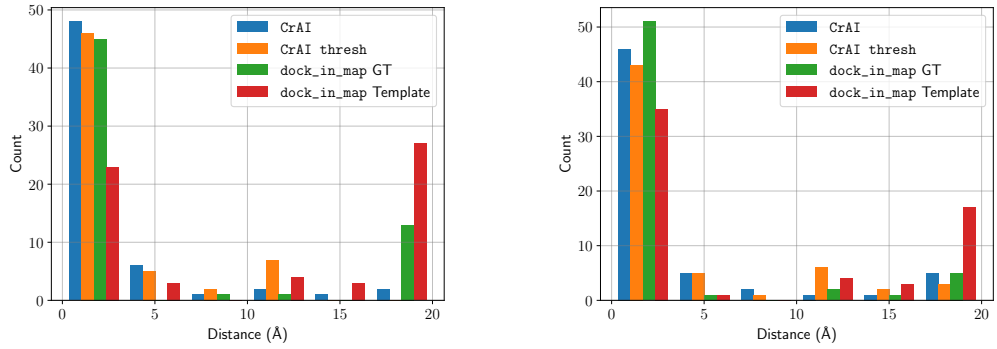


Figure S.2: Capped histogram of distances between predictions and experimental structures on Fabs. `random` is on the left and `sorted` on the right.

As can be seen, our tool widely outperforms `dock_in_map`. Under the hit rate metric, we have the following performances (by order of tools) : `random` : 94.1, 88.8, 62.2, 19.4 `sorted` : 97.3, 93.9, 71.1, 8.5.

B.2 Dependence in resolution

We concatenate the predictions for all our systems and scatter them as a function of the resolution of the maps. We cap the predicted distances to 10Å and add a best fit curve. We show the result in Figure S.3. We see a moderate dependency in resolution, and many successful ones for resolutions above 4. In contrast, `dock_in_map` exhibits a strong dependency to resolution.

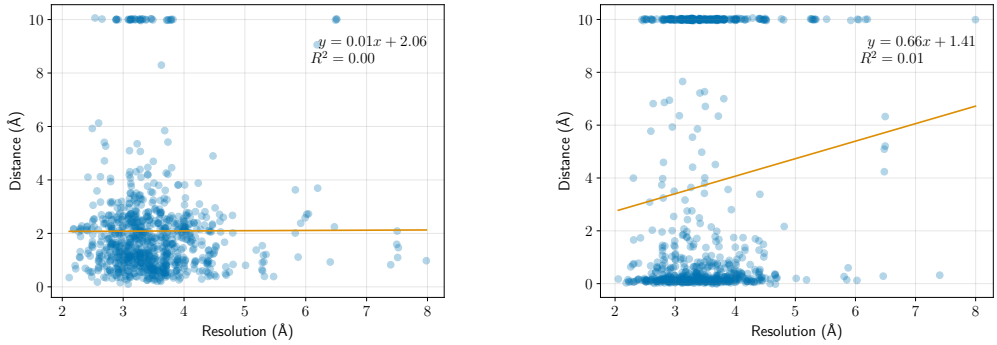
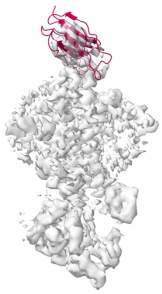
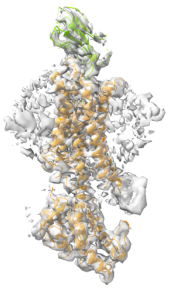
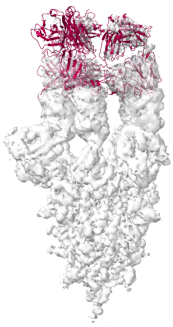
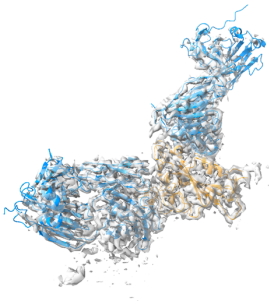
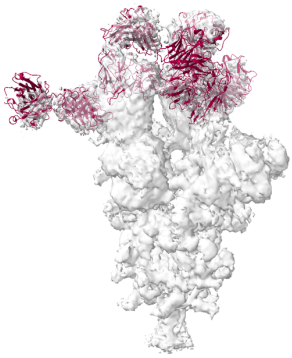
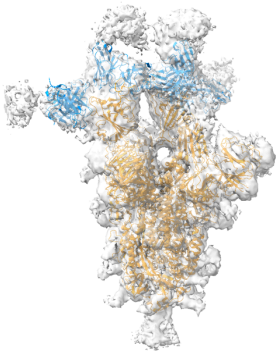
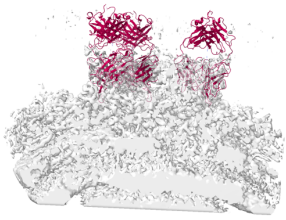
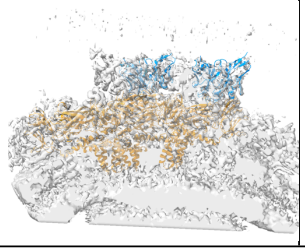
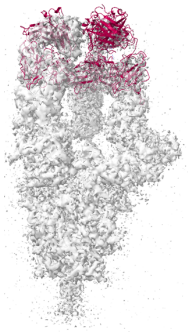
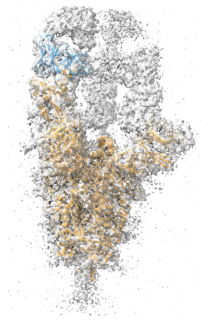
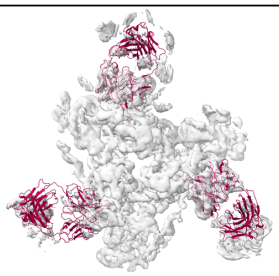
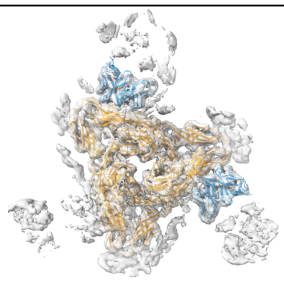
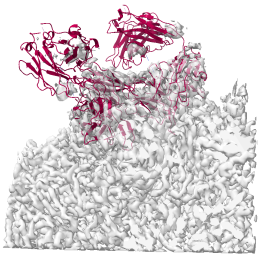
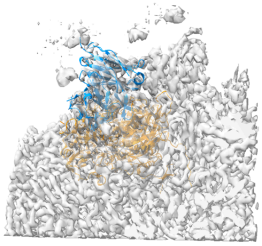
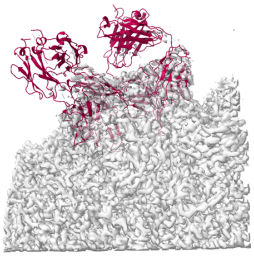
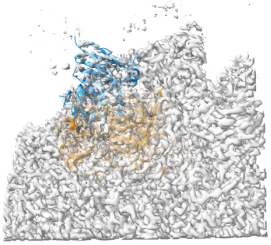


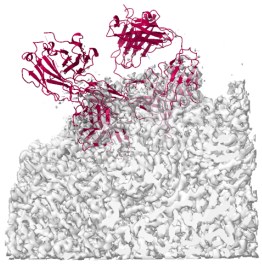
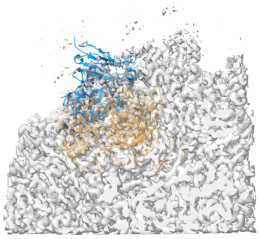
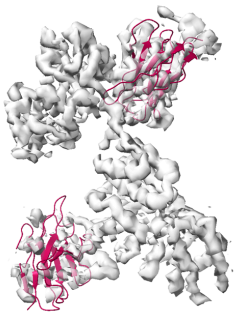
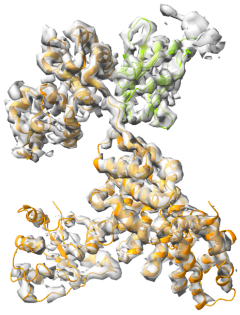
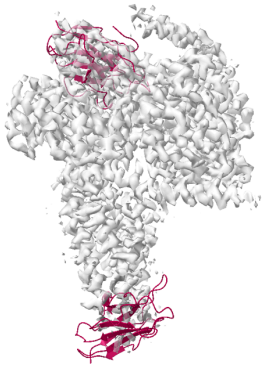
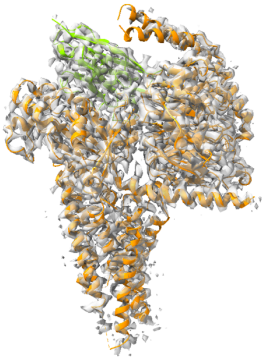
Figure S.3: Scatter plot of the performance of `CrAI` (left) and `dock_in_map` (right) as a function of the resolution. x-axis is the resolution of the maps. y-axis is the distance between a prediction and its corresponding system, capped to 10Å for failed systems.

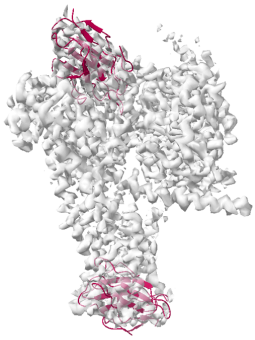
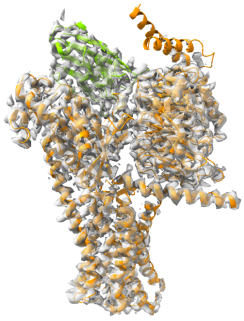
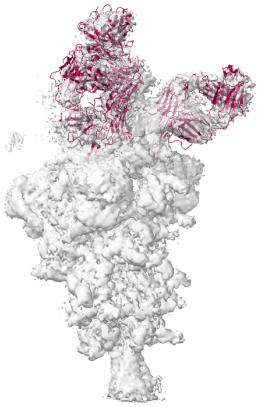
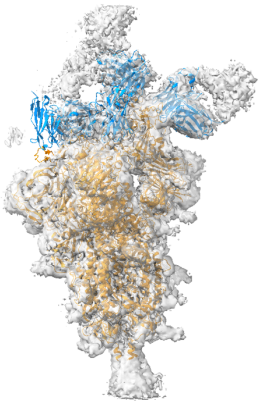
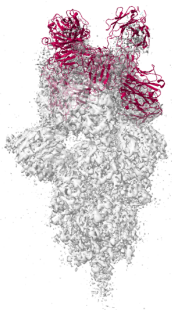
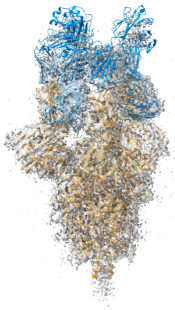
B.3 Failed examples

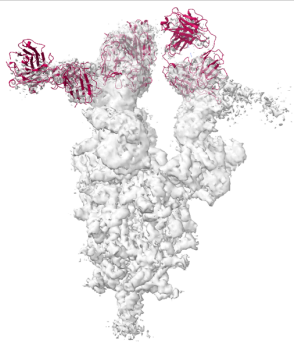
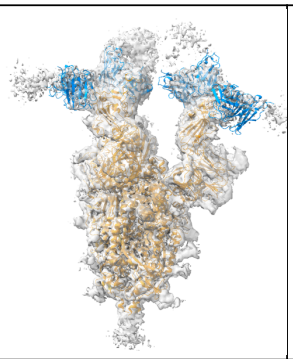
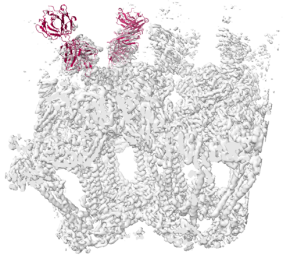
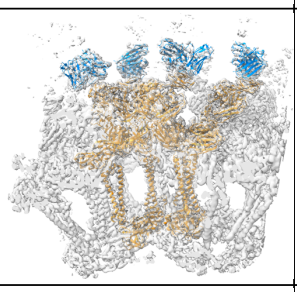
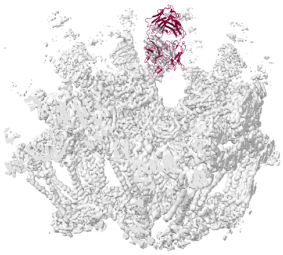
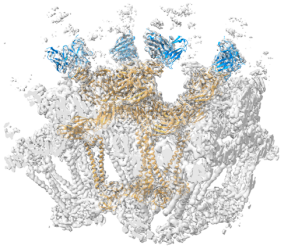
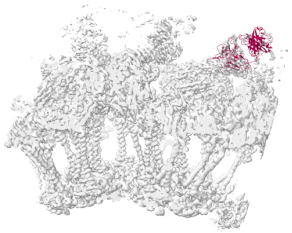
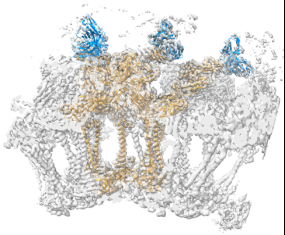
PDB file	Predicted PDB	Ground truth	Details of the error
Metrics computation was wrong			
7YMH			Not an error
7YVN			Not an error
Too many objects			
7XDA			1 Fab more No density visible for this third Fab -

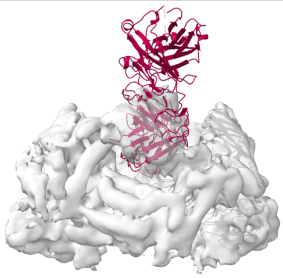
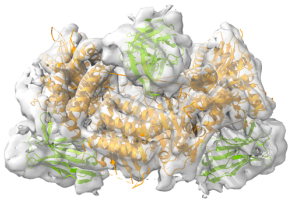
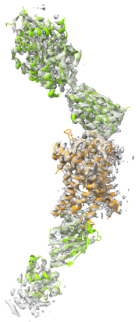
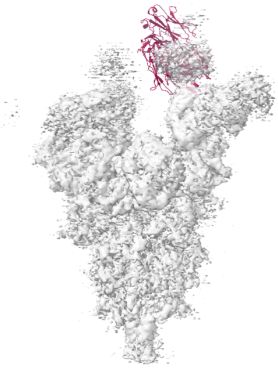
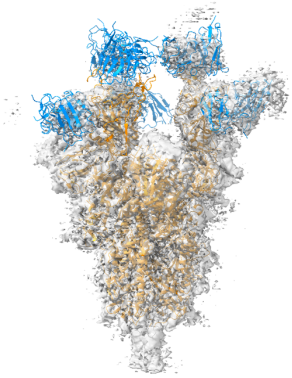
7YM8			1 more Pretty bad as it fit a nanobody in a helical domain
7YVP			1 Fab more Not a very clear density for the extra Fab
8CXI			1 more but placed correctly

8D0Z			2 more but correctly placed in the density
8DEF			1 more correctly placed based on the symmetry but for which the density is weak.
8E8L			1 more correctly placed in the density
8E8R			1 more correctly placed in the density

8E8X			1 more correctly placed in the density
8GNJ			1 more
8GW8			1 more

8HMP			1 more – threshold Dependant – top Nanobody recognized correctly
Missing objects			
7XJ6			1 less Bad density for the 4th Fab
7YC5			1 less

7YVO			1 less Bad density for the 4th Fab
8DWW			**2 less**
8DWX			3 less
8DWY			3 less

8EQB			**3 less – Fab in place of a nanobody**
8G8W			1 less nanobody – it's special because they are megabodies
8H07			2 less with very Bad density

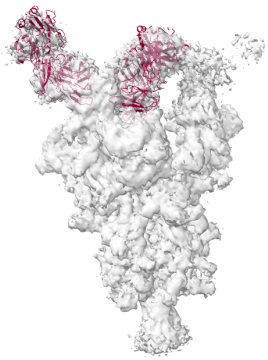
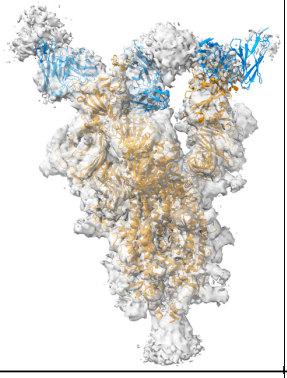
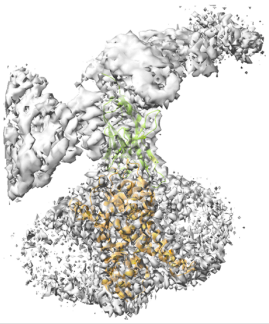
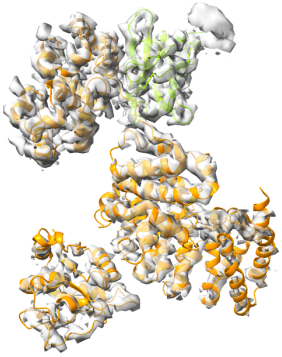
8HEC			1 less with bad density
8J1N			1 less nanobody 1 more Fab (but there is a density that seems like a Fab)
Wrong positioning			
8GNI			**Wrong position in a helical region!**

Figure S.3: All failures, split by error type