Building molecular model series from heterogeneous CryoEM structures using Gaussian mixture models and deep neural networks

Muyuan Chen[1*]

[1] Division of CryoEM and Bioimaging, SSRL, SLAC National Accelerator Laboratory, Stanford University, Menlo Park, CA 94025, USA

* Correspondence to: muyuanc@stanford.edu

Abstract

Cryogenic electron microscopy (CryoEM) produces structures of macromolecules at near-atomic resolution. However, building molecular models with good stereochemical geometry from those structures can be challenging and time-consuming, especially when many structures are obtained from datasets with conformational heterogeneity. Here we present a model refinement protocol that automatically generates series of molecular models from CryoEM datasets, which describe the dynamics of the macromolecular system and have near-perfect geometry scores.

Main

Single particle cryogenic electron microscopy (CryoEM) has emerged as the leading technique for determining the structures of proteins and RNAs in recent years[1–3]. Instead of producing one static 3D structure per sample, CryoEM provides the opportunity to explore the dynamics of the macromolecular systems. The *in silico* classification techniques enable the determination of multiple states from a single dataset, revealing the assembly steps of a macromolecular complex[4,5], or functioning mechanisms of membrane channels[6,7]. With advanced machine learning methods, particles can be mapped to a latent space that describes the conformational landscape of the system and provides insights into the continuous movement of different domains of macromolecules[8–11].

As our capability of heterogeneity analysis expands, interpreting the conformational landscape of a macromolecular system and building molecular models for the many CryoEM maps produced by the analysis becomes a major bottleneck. For a single high-resolution structure, an atomic model can be built by refining homolog or predicted models against the map[12,13], or using the *de novo* modeling tools directly from the 3D reconstruction[14,15]. While existing tools can automatically produce reasonable starting models, it is often time-consuming to refine them and get final models that fit well into the maps and have good stereochemical geometry. In addition to running automated refinement software[16,17], the modeling process frequently involves multiple iterations of semi-manual model adjustment[18], as well as repeated submission to the PDB validation server[19] to achieve satisfying validation scores.

When a system involves continuous conformational changes, the modeling task becomes more challenging. To describe a movement trajectory, one strategy is to build one model at each

distinct conformation and morph between the models by interpolation[20,21]. However, simple interpolation does not necessarily capture the transition between the conformational states, and the intermediate models along the morphing trajectories are not guaranteed to have the correct stereochemical geometry. To truly model the transition between conformational states and describe the dynamics of macromolecular systems, new modeling approaches are needed to build an ensemble of molecular models from particles of different conformations, using the structural variability information captured by heterogeneity analysis techniques.

Previously, we showed the Gaussian mixture model (GMM)-based protein structure representation, combined with deep neural networks (DNN), can be used to extract structural variability information from CryoEM datasets[10,22]. Here, we present a computational method, similarly based on the GMM-DNN architecture, that builds a series of molecular models from a CryoEM dataset with structural variability. Each model in the ensemble would satisfy protein geometry constraints and fit to the reconstruction of particles from the same conformation at the target resolution.

The modeling process starts from an existing homolog or predicted model, where we place one Gaussian function at each non-hydrogen atom. Using a DNN, we first refine the GMM, so it fits the consensus 3D reconstruction of all particles. Then, taking the existing result from the heterogeneity analysis of a CryoEM dataset, the algorithm produces a series of atomic models, each corresponding to a conformation from the same point of the latent space.

During the refinement, the map-model Fourier shell correlation (FSC) is used to guide the model fitting, and a cutoff frequency is set to avoid over-interpreting the high-resolution information in the map. In contrast to existing real-space refinement methods[16,17,23], the properties of the GMM allow the algorithm to optimize the coordinates of atoms in the real space, while evaluating map-model similarity in the Fourier space. In addition to the FSC, we also introduce stereochemical constraints of proteins and RNAs to the loss function during DNN training, which ensures that the model at every frame along the trajectory has valid geometry. To enable gradient-based optimization for DNNs, we re-implemented the empirical constraints, including Ramachandran plots and sidechain rotamer libraries[24], as differentiable forms. The DNN-based optimization makes it possible to converge to a globally favorable geometry faster without human interference.

To demonstrate the protocol, we start with a classical modeling task, where one atomic model is refined against one high-resolution CryoEM structure. In this example, we refine an existing model of TRPV1 (PDB-3J5R)[25], built from a lower resolution structure, and fit it to a higher resolution map (EMD-8117, 2.95Å)[26]. This is performed through an automatic multi-step refinement process, using three DNNs that capture structural information of different resolutions. The process involves large-scale morphing, residue-wise adjustment, and finally full atom refinement that considers both the map-model agreement and the stereochemical constraints (Figure S1). The resulting model shows better real space fitting, higher Q-score[27], and near-perfect PDB validation metrics (Figure 1A-D).

To show the robustness of the method, we applied it to 25 recent CryoEM structures from the PDB/EMDB, including proteins and RNAs and ranging from 2.5 to 5.7Å resolution. Fully automated refinement with default parameters yielded better geometry scores for every example, while the Q-score is largely unchanged (Figure 1E, S6). This indicates that our method improves the overall geometry of the molecular models without sacrificing the map-model agreement.

Now the refinement protocol of single molecular models is established, we apply the method to refine a series of models from the previous heterogeneity analysis of CryoEM data. We again start from the TRPV1 example (EMPIAR-10059), from which the movement of Ankyrin repeat domains has been observed previously[11,22]. Starting from the heterogeneity analysis results, we follow a 1D trajectory in the conformation space, group particles by regular intervals, and generate 3D reconstruction from each group of particles. Then, given the positions of the particle groups in the conformational space as input, DNNs are trained to output molecular models that match the corresponding maps of the particle groups, resulting in a series of models that describe the conformational change captured by the heterogeneity analysis. In addition to the map-model similarity, the stereochemical constraints are also considered during the DNN training, so the output model from any conformational point would have minimal clashing and good geometry (Figure S2, Supplementary video 1).

Finally, we use the spliceosome dataset (EMPIAR-10180) as an example to demonstrate the capability of the protocol[28]. The spliceosome is a complex of more than 100,000 atoms, including both protein and RNA, and the large-scale structural flexibility of the system has been well documented[9,10,29,30]. Using our method, we build molecular models that describe the continuous movement within the system, which match the results of existing heterogeneity analysis. Moreover, each snapshot model along each movement trajectory has few atomic collisions and near-perfect geometry score (Figure 2, Supplementary video 2).

In sum, for single model refinement tasks, our atomic model refinement protocol produces better models than results from existing methods, according to the PDB validation metrics. While a perfect geometry score does not necessarily guarantee a good model, the flexible nature of the DNN makes it straightforward to adopt any new validation criteria in the future. The GMM-based architecture also connects the modeling step to the heterogeneity analysis of CryoEM datasets seamlessly, making it possible to build a series of models that describe the structural dynamics of the macromolecular system. The entire process is performed automatically without any manual intervention. Compared to stacks of 3D reconstructions, the molecular model series makes it easier to interpret the dynamics of the protein complex and provides a convenient way to compare the structure flexibility information obtained by CryoEM with results from other techniques, including MD simulation and NMR[31–33].

Software availability

All computational tools described here are implemented in EMAN2, a free and open source software for CryoEM/CryoET imaging processing. The code is available at github.com/cryoem/eman2, and a tutorial can be found through eman2.org/e2gmm_model.

Data accessibility

All data used in the paper are publicly available through EMPIAR, EMDB and PDB. Models produced in this paper will be deposited in the PDB upon paper acceptance.

Acknowledgements

This research is supported by NIH grant R01GM150905.

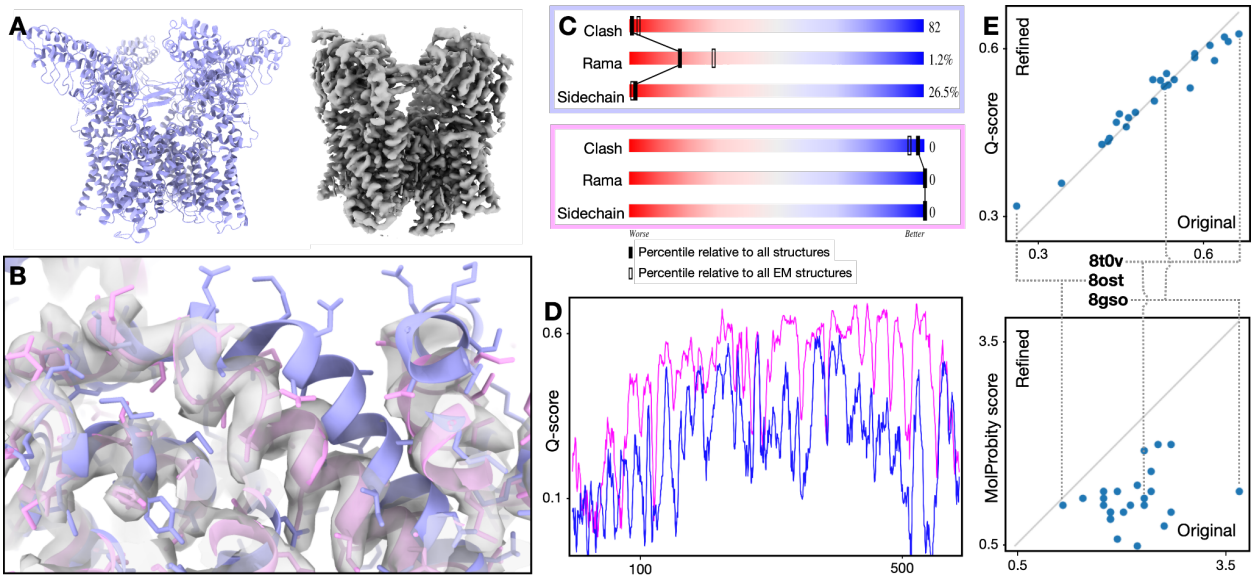

Figure 1. Single model refinement. (A) Input model (PDB-3J5R) and CryoEM map (EMD-8117) of TRPV1. **(B)** Zoomed-in view of map and models. Blue - input model; pink - GMM refined model. **(C)** PDB validation results of the input (top) and refined (bottom) model. **(D)** Q-score comparison of the input (blue) and refined (pink) model. **(E)** Comparison of the original and refined version of 25 models from the PDB. Top - Q score; bottom - MolProbity score (lower score indicates better geometry).

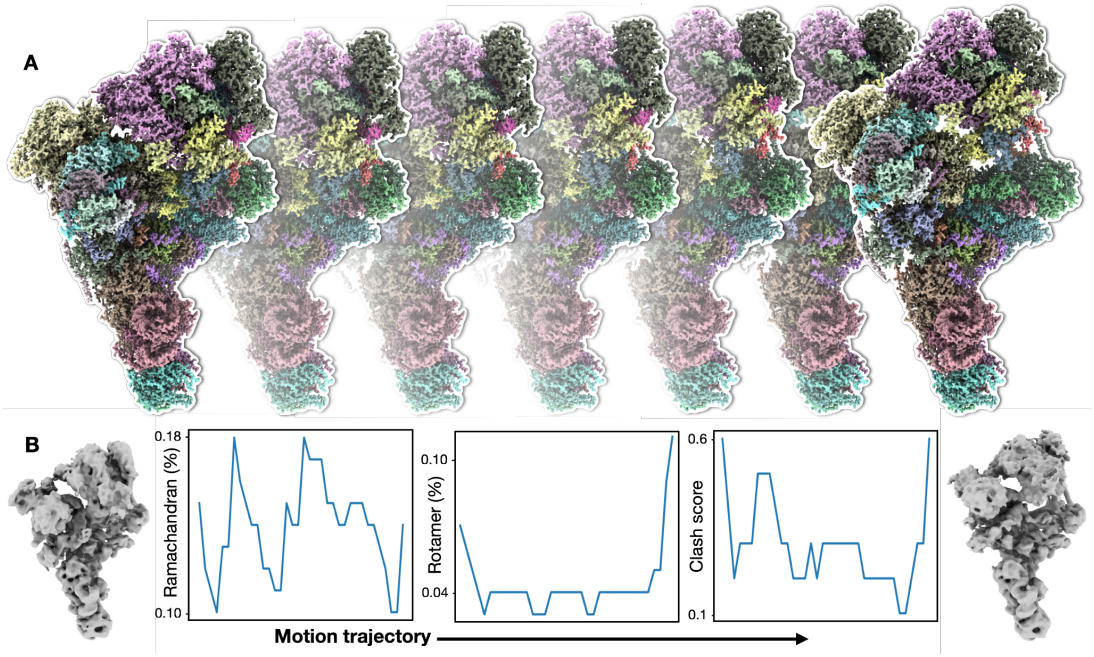

Figure 2. Model series refinement. (A) Snapshots of GMM refined molecular model series along one continuous motion trajectory of spliceosome from dataset EMPIAR-10180. **(B)** Left and right panels - 3D reconstruction of the first and last frame of the motion; middle three panels - Ramachandran, rotamer outlier, and clash score for models along the motion trajectory.

Methods

GMM for atomic model representation

The basic concepts of GMMs and the architecture of the DNNs have been described in our previous publications. In this section, we briefly explain the basic concepts of the method, but it is recommended to refer to the previous papers for more details[10,22,34].

To represent the structure of proteins, we use a GMM, which is a sum of many Gaussian functions in real space, with each Gaussian function placed at one non-H atom. Each Gaussian function is represented by five parameters: amplitude, width, and the 3D center coordinates. Similar to our previous GMM-DNN based heterogeneity analysis, to reduce computational resource consumption, the DNN is trained on batches of 2D projection images instead of 3D volumes. To compare projections of the GMM and projections of the map, we use the Fourier ring correlation (FRC) as the loss function. The FRC between the Fourier transform of two images is the average of the correlation coefficients over Fourier rings. Since each ring is independently normalized, the FRC is insensitive to filtering of images, or the sharpening methods applied to the reconstructions.

Basic stereochemical constraints

We first consider the length and angle of the covalent bonds in the molecule. Initially, we take the mean and standard deviation (std) of each type of bond for every amino acid residue from AlphaFold[12]. However, when refining models with those parameters, we notice that there is an inconsistency between the AlphaFold values, and the statistics used in the PDB validation server. For some types of bonds, smaller std values are used in the PDB validation, so outliers become more likely when refining with those constraints. To compensate for this, we gathered information from many PDB validation reports and corrected those values manually, so the statistics agreed with PDB validation.

In reports from the PDB validation server, bonds with lengths that are >5 std from the average bond length value of that type are considered outliers and receive a penalty in the validation score. To enforce the ideal bond length and angle, during DNN training, we included both the likelihood that the current bond length and angle follow the Gaussian distribution of the target mean and std, as well as an additional penalty for outliers. Here we set the threshold for outliers to be 4.5 std by default, slightly tighter than the PDB validation server requirement, to overcome the potential error caused by the PDB file format, as well as any remaining discrepancy between the ideal bond values used by us and the validation server.

In addition to the bond length and angles, we also compute dihedral angles from the model. Here, we first apply the constraints for the planar dihedral angles, which enforce sets of atoms to always sit on the same plane. Keeping consistent with the PDB validation server, the threshold of the maximum allowed peptide bonds dihedral angle is set to be 30 degrees, and

the threshold for other planar bonds, including amino acid backbone and some planar sidechains, to be 10 degrees.

During the DNN training, all bonds and angles are directly computed from the GMM, using the center of Gaussian functions as the coordinates of the atom. To speed up the process, before the refinement, we pre-compile a list of atom pairs that form bonds, as well as the ideal bond length and std for each bond on the bond type. During the refinement, for each batch, the program evaluates the length of all the bonds in the list and compares them to the ideal values. Similarly, we pre-compile the 3-atom lists for bond angle, and 4-atom lists for dihedral angle calculation so the geometry can be quickly evaluated during DNN training.

Ramachandran plot

Ramachandran plot describes the preferred geometry of peptide chains using dihedral angles along the protein backbone. With two angles per residue, it is essentially a 2D histogram with finite sampling (often at 1 degree). While it is straightforward to compute a Ramachandran score from any protein backbone, it is challenging to use the score as a part of the loss function during the DNN training, because the analog gradient, which is required by most DNN optimizers[35], cannot be computed from the discrete histogram. To overcome this, we need to convert the Ramachandran plot into a differentiable form. Here, similar to our 3D GMM-based protein density representation, we fit 2D GMMs to the Ramachandran plots, turning the discrete histogram into a continuous function (Figure S3).

For the GMM representation of Ramachandran plots, we use the six 2D histograms for the different residue types defined in Phenix[36]: General, Gly, trans-Pro, cis-Pro, pre-Pro, and Ile. For each type, 2500 Gaussian functions are used to fit the histogram. Because the threshold that determines an outlier in the Ramachandran plot in PDB validation is very small (0.0005, while the highest value in the histogram is 1), even when the GMM fitting is good, there can still be many Ramachandran angles that are considered acceptable by the GMM representation but are outliers in the histograms. Since Ramachandran outlier count is the main metric used by the PDB validation to evaluate the protein backbone, here we fit the logarithm of Ramachandran score ($ln\ (R + C)$, where *R* is the original Ramachandran score, and *C* is a small constant set to e^{-10}). We also introduce additional penalties along the outlier boundaries of the plot during the GMM fitting, ensuring all Ramachandran angles that are considered outliers from the discrete histogram are also considered outliers in the GMM representation. Similar to the GMM representation of protein structures, the fitting of 2D GMM for Ramachandran plots is also done by densely connected neural networks. The RMSD of the GMM fitting is smaller than 0.001, and virtually all outliers in the histograms are outliers in the GMM representation.

With the GMM-based differentiable Ramachandran plot, we can then compute the Ramachandran score from the dihedral angles of any given model. Similar to the metrics used in PDB validation, here we only include the percentage of “allowed” and “outlier” residues as a part of the loss function during the DNN training. With the smooth GMM representation of the

Ramachandran plot, the optimizer can compute the gradient of the score with respect to the atom coordinates automatically for any atomic model.

Sidechain rotamers

Similar to the Ramachandran score, the preferred sidechain rotamers are also defined as discrete histograms of the Chi angles of the amino acid sidechain. So similarly, we fit GMMs to those histograms in order to have a differentiable loss function (Figure S4). Here, we use the rotamer library from Phenix. The number of Chi angles varies from 0 (Gly) to 4 (Arg, Lys) depending on the type of amino acid, so the dimension of the GMMs also changes accordingly. For all the amino acid types, the preferred rotamer histogram consists of a few Gaussian-like peaks on a near-zero background, making GMM fitting relatively simple. For sidechains with 1, 2, 3, and 4 Chi angles, we use 4, 25, 64, and 160 Gaussian functions respectively to fit the rotamer histograms. Since there are only a few peaks in the rotamer space, to fit the GMM, we simply place the center of Gaussian functions at those peaks and optimize the GMM parameters locally. Similar to the Ramachandran fitting, to increase the weights of small value outliers, here we fit the logarithm of Rotamer scores instead of the original values. After generating a GMM for each type of amino acid side chain (excluding Gly), the gradient of the rotamer score with respect to the coordinates of atoms can also be computed by automatic differentiation. Similar to the bond length and angle constraints, for sidechain rotamers, we consider both the likelihood that the Chi angles follow the Gaussian distribution, and the additional penalties for rotamer outliers.

In addition to including the rotamer constraints as a part of the model refinement, we also offer an option to fully rebuild the rotamers from a given CryoEM map. To do this, we enumerate all peaks in the GMM of the sidechain rotamer for each residue and pick a rotamer with the best map model similarity. Afterward, the Chi angles of all rotamers are refined locally to maximize both the map and the geometry score. This functionality is particularly useful when fitting models built from low-resolution maps, where the sidechain rotamers are unreliable, into high-resolution maps with clear sidechain densities. In those situations, it can be difficult to converge to a different rotamer that agrees with the map from the starting point, because good rotamers only form isolated peaks in the space of Chi angles.

Avoiding clashing atoms

Clashing of atoms in a molecular model is detected when the distance between two unlinked atoms is smaller than the sum of their van der Waals (VdW) radius, minus a certain threshold value (0.4Å in PDB validation). Initially, we used the VdW radius of each atom in each amino acid type from AlphaFold. However, like in the case of bond length/angle, there turned out to be a discrepancy between the VdW radius from AlphaFold, and those used in the PDB validation server. It is also notable that the VdW radius values used in the PDB validation server are also different from those used in Phenix, which further complicates the process. To obtain the precise VdW radius values for all atom types from the PDB validation server, we upload multiple synthetic atomic models to the server, which includes clashing atom pairs of all atom types from

every amino acid residue. The VdW radius of the different atom types is then directly parsed from the validation report and used in the refinement protocol.

To calculate the atom clashing score during model refinement, for each atom in the model, we track a number of neighboring atoms (default 128) that it might clash with, since computing an all-vs-all contact map at every iteration would be too expensive. The neighboring atoms are selected using the KD-Tree algorithm at the beginning of the refinement, and any atom pairs that are directly bonded, or indirectly connected within 3 bonds are excluded. The neighbor list is updated every 100 iterations of refinement, in case faraway atoms start to form contact as the models morph during refinement.

Another major issue when calculating clashing atoms is the introduction of H atoms. As described earlier, H atoms are not included in the GMM, since they are generally invisible in CryoEM maps before reaching true atomic resolutions[37]. However, since H atoms are used in the PDB validation server for clashing score calculation, we have to add them back to the model to optimize for clashing. The H positions are based on the coordinates of the non-H atoms of the corresponding residue, and the exact relative positions of H atoms in all amino acid types are adapted from Phenix. To simplify the process, the DNN outputs models of only non-H atoms, and the H atoms are added to that model as a post-processing step during training.

There are two additional points of concern when computing the clash between H and other atoms. First, when a potential H-bond can form between two atoms, we increase the threshold that defines clashing between the two atoms by 0.4Å, i.e. the atoms can stay closer without being considered clashing. Here H-bonds are simply defined as a pair of H and O/N atoms that are not covalently bonded, and the angle of H-bonds is not considered.

Second, while in theory some of the H atoms, such as those in -CH3 or -OH, have an extra degree of freedom and can rotate around the bond, strangely, it turned out that the PDB validation server does not take this into consideration when computing clash scores even though the validation report claims that the H atoms are optimized. When the uploaded model includes H atoms, the validation server will remove all H atoms, before adding the H atoms back at their default zero-degree torsion angles. Therefore, using optimized non-zero torsion angles for H atoms will lead to higher clash scores from the validation server, even if those clashes can be resolved by the rotation of H atoms. So, we forbid H rotation during the refinement and only place H atoms at the zero-degree conformation defined in the PDB validation server.

RNA score

In the PDB validation, the RNA score is defined as the likelihood that the backbone of each RNA base belongs to one of the 46 suites[38]. Seven dihedral angles are computed from the backbone of each RNA base, and the suite score is calculated from the distance from the 7-angle vector of that base to the nearest pre-defined suite cluster center in the 7D space. Compared to the Ramachandran plot or the sidechain rotamers, the computation of the RNA score from the validation report is less intuitive. Specifically, RNA bases that do not belong to

any suite (score smaller than 0.001 for any of the 46 clusters) are ignored in the final score, which is an average of the score from all bases that have an assigned suite. As a result, having outliers in RNA backbone geometry does not actually lead to a lower score, and each non-outlier RNA base needs to get as close to the corresponding cluster center as possible to achieve a higher score.

Although the metric used in the validation does not consider outliers, during the refinement, we still assign each RNA base to the nearest suite center in the 7D space, without skipping bases with low scores. As such, an outlier RNA base with a near-zero score would still contribute to the overall RNA score. This prevents the DNN from converging to obviously incorrect high-score solutions, such as having only one base with a perfect score and everyone else an outlier.

DNN-based single model refinement

Four-layer densely connected neural networks are used for the model refinement. The network structure is the same as the decoders used in heterogeneity analysis[10], which takes a conformation input and outputs the GMM parameters. Three neural networks are used for the refinement: one for the initial large-scale morphing, one for residue/base wise refinement, and another for the full atom refinement. The outputs of the three neural networks are added together, all on top of the GMM built from the input model, so they can collaborate through a boosting-like mechanism. Starting from the long-range morphing, each network can refine the finer details of the model, target higher resolution, and consider more stereochemical constraints. The hierarchical design of the protocol ensures refinement can capture both the large-scale conformation change and the subtle geometry improvement, compared to the existing model refinement tool like Phenix (Figure S5).

Similar to our GMM-DNN based heterogeneity analysis, to reduce resource consumption, the DNN is trained on batches of 2D projection images instead of 3D volumes. Since there is only one reference map in the single model refinement, the input to the DNN is always a vector (4D by default) of constant ones. For each projection image in the batch, the DNN generates one GMM that includes all non-H atoms in the model. The GMM is projected to 2D, and FRC between the GMM projection and the projection of 3D volume is computed, which is used as the score for map-model similarity. From the GMM, various geometry scores can also be calculated, including the deviation of bond length and angle, the Ramachandran and rotamer scores, as well as penalties for clashing atoms.

For the model refinement, one major difficulty is to balance the weight between the score of map-model similarity and the score of the model geometry. To address this, an automated method is used to determine the weighting factor from the data itself. We start by running two iterations of the refinement that minimize the map-model similarity score without the model geometry constraints. From the refinement, we calculate the ratio between the increase of the map-model similarity score and decrease of the geometry score over the two iterations. This ratio quantifies how much model geometry quality is sacrificed to achieve the improvement of map fitting. Then, we set the weight of the geometry score to the reciprocal of that ratio and re-

initialize the DNNs to restart the training. As a result, in the new iterations, the map-model similarity score and the geometry score will be roughly balanced, i.e., the geometry score changes at a similar scale as the map-model similarity during the training.

Generally, the refinement process includes five steps (Figure S1). First, we pre-compile the geometry information from the model, so that all the geometry scores can be directly computed later as a part of the DNN training process. At this step, the input model file is parsed and the information of all the bonds, angles and dihedral angles, as well as their ideal values, are converted into matrix forms so the corresponding geometry scores can be calculated during the refinement.

Second, we train the first DNN to deal with the potential large-scale domain movement between the input model and the given CryoEM map. To model this, we use the hierarchical GMM architecture we previously developed[22], and divide the full atomic model into a small number of patches (64 by default). To maximize the map-model similarity, the DNN is trained to output the transform of each patch, which is then applied to the full GMM and morphs the atomic model. Here, the patches are divided automatically from the centers of all amino acid residues (or DNA/RNA bases) using K-means clustering, so atoms of the same residue/base are always in the same patch. This forces each residue/base to move as a group during the large-scale morphing and avoids introducing additional geometry errors within residues.

In the third step, another DNN is introduced to adjust the model at the residue/base level. The second DNN is implemented using a similar approach as the first one, except each patch contains only atoms from one residue. At this step, the two DNNs are trained together, and their outputs are summarized so the second DNN can perform residue-level geometry adjustment on top of the large-scale domain movement learned by the first DNN in the previous step. Also at this point, we start to introduce the basic geometry constraints including bond length, angle, and clash score into the loss function.

After the residue-wise refinement, we next optimize the sidechain rotamers. For high-resolution maps, as described above, we rebuild the rotamers for all residues first. Then the rotation of each sidechain Chi angles is optimized to better fit into the map while also keeping an acceptable rotamer score and not clashing with other atoms. Since this is only a local search, it is done without the DNN, by directly optimizing the Chi angles with the Adam optimizer[35].

Finally, we refine the full atom model with the third DNN, taking all the stereochemical constraints into consideration. This DNN generates the full GMM that includes all non-H atoms, and outputs from three DNNs, as well as the side chain torsion angles from the previous step, are summarized together to produce the final model. The final step can also run independently without an input map, to only refine the geometry of the model locally, without the constraints from structure data.

Continuous model series refinement

The input of the continuous model series refinement is a stack of 3D reconstructions, along with their corresponding assigned conformation values (i.e. location in the latent conformational space during the heterogeneity analysis). So results from any continuous heterogeneity analysis protocol[9,11,30], in addition to our GMM-DNN based one[10], can be used as the input for the modeling process. Same as the single model refinement, we make projections from the reconstructions, and use those 2D projections instead of 3D volumes as DNN training input to reduce the computational resource usage.

The procedure of model series refinement is generally the same as the single model refinement, but instead of a constant vector, here for each projection image, the actual conformation value of the corresponding reconstruction is used as the input of the DNN. To ensure the continuity of the output models, a small random variable is added to the conformation input, similar to the variational autoencoder implementation[39]. Therefore, the output models can have continuous movement even though the input reconstructions are only sampled at discrete positions in the conformational space.

Finally, to make sure the model series always has good geometry, we perform additional rounds of local geometry refinement using the input of uniformly distributed random variables along the trajectory in the latent space. As a result, the output model series would have near perfect stereochemical scores even at the frames that are not sampled by the 3D volumes from the input.

Details on examples

For the TRPV1 example, we start from the PDB model (PDB-3J5R)[25], which was built from a lower resolution structure at a different conformation (EMD-5777, 4.2 Å). The model was first inspected and roughly fitted into the new map (EMD-8117, 2.95 Å)[26] using UCSF ChimeraX[40], before being used as the input for the refinement protocol. The target resolution for model refinement was set to 3 Å, and default parameters were used for the refinement. Since the map resolution is high enough, we also rebuilt all sidechain rotamers according to the density map during the refinement.

The continuous movement of TRPV1 was generated from the public dataset EMPIAR-10059, using the heterogeneity analysis previously described[22]. The heterogeneous model series was built from a series of 3D reconstructions generated along the first eigenvector of the particle distribution in the conformation latent space, which describes a rotation motion of the ankyrin repeats domain. The output model from the previous single model refinement of TRPV1 is used as the input for the model series refinement. The target resolution of the model series refinement was set to 7Å, and default parameters were used during the process (Supplementary video 1).

To model the continuous movement of the spliceosome, we use the dataset EMPIAR-10180[28], and heterogeneity analysis results previously described. The original model (PDB-5NRL) was used as the input for model refinement, and the 3D reconstruction from the neutral state of the

heterogeneity analysis was used as the target map for the initial single model refinement[10]. A series of 3D reconstructions were generated along one eigenvector in the conformation latent space, and the target resolution of the model series refinement is 15Å (Supplementary video 2). Additionally, movement along a circular trajectory is also modeled (Supplementary video 3). Default parameters were used during the refinement, except that the batch size was set to 2 in order to fit the large complex in GPU memory.

To demonstrate the performance of the model refinement protocol in general cases, we picked 20 protein structures and 5 RNA structures from the PDB. To show that our method can improve structures with relatively poor geometry, we select models with validation scores at the lower 50% percentile using the advanced search function provided by PDBe[41]. To make sure the selected models were generated using the latest model refinement techniques after the PDB validation service was established, we sorted the PDB by deposition date, and picked the 5 latest deposited protein models (at the time of search) for every 10 validation score percentile. I.e., we include 5 newest models with validation scores between 0-10%, 5 models with scores between 10-20%, and so on. The 5 RNA models were selected by sorting the models, which include RNA and have validation scores between 0-50%, by deposit date in PDBe. Some models from identical publications, as well as models associated with partial or unaligned CryoEM maps, were excluded from the process. From their original publication, the models were generated with both automatic refinement and manual adjustment, and the software used included Phenix, Coot, and NAMD[16,18,42]. The PDB ID of the models are: 8t0q, 8gso, 8wyc, 8unh, 8xom, 8wx0, 8t0v, 8q74, 8xqp, 9c49, 8xzb, 9c57, 8k7t, 8xse, 8wly, 8xqr, 8k03, 8xqa, 8k3j, 8k11, 8ost, 9enf, 8tjv, 7yg9, 8uau.

For each of the 25 models, default GMM-based single model refinement was performed, using the PDB model and the corresponding CryoEM maps from EMDB as input. The geometry scores of both the original and refined models were calculated using MolProbity (which the PDB validation server is mostly based on) provided through Phenix; the RNA suite scores were computed using the Suitename tool from Phenix; and the Q-score was calculated using the plugin in UCSF Chimera[24,27,36,38,43].

From the plot in Figure 1E, the Q-scores of the models remain relatively constant after refinement, while the protein geometry score, as well as RNA scores, improves in every case (Figure 1E, Figure S6. Note that better geometry is indicated by lower MolProbity and higher RNA suite score). This is expected since unlike the TRPV1 example shown in Figure 1A-D, in which we fit a model into a higher quality map of different conformation, here each model was built originally from the associated map. At a closer look, it is worth noting that for most of the entries, there is a small increase of Q-scores after the refinement, but for the few models with the highest initial Q-scores, our model refinement decreases the score slightly. This is because many sidechains in those models were tightly fitted into the map, without the concern of “allowed” rotamers. Since our refinement protocol finds the acceptable rotamers that fit best to the map, it led to much fewer rotamer outliers, but slightly worse sidechain fitting.

Resolution for model refinement

Conceptually, it is easy to assume models built from lower-resolution structures would have lower geometry scores. However, since we use FSC as the metric for map-model similarity, the model only needs to agree with the map at the target resolution, and the geometry of the model can be further refined without impacting the map-model FSC. Theoretically, assuming we have a perfect estimation of map resolution and model geometry, for a map determined at any resolution, there is at least one model that fits the map at the determined resolution, which also has the ideal geometry. For example, given a 5Å CryoEM map, it should be possible to build a model with a perfect validation score that also agrees with the map. Without the high-resolution information, it is impossible to confirm the refined model is exactly the correct one, i.e., it will match the map precisely if the structure can later be determined at 3Å or higher resolution. However, it is still a plausible model given the information from the map, as well as the prior information we have about protein geometry. So, it is safe to argue that such a model is clearly better than alternative models that also agree with the map at 5Å resolution but with poor geometry scores. Additionally, since in many cases, the resolution of CryoEM structures is limited by the heterogeneity within the system, it is more likely that there are in fact many models with good geometry that agree with the map at 5Å resolution, each of them being equally correct as each particle can adopt one of many conformations.

Using the GMM-based representation, we can refine the atomic coordinates in real space, while evaluating the map-model similarity in the Fourier space. This makes it possible to explore the different models that match the 3D reconstructions equally well at the target resolution and find the one with the best possible geometry. Despite the capability, similar to all CryoEM practices, it is still risky to interpret the structures beyond that resolution, especially for small features such as the opening size of a protein channel. For single model refinement, since the resolution of the map is estimated by the gold-standard FSC curve, it is relatively easy to set that as the target resolution for the model refinement. The issue is more complicated for the model series refinement, because there has not been an established method that estimates the resolution of the continuous movement trajectories resulting from heterogeneity analysis of CryoEM data. In the model series refinement examples presented in the paper, we decide the target resolution by visual inspection of the individual 3D reconstructions. This is likely an underestimation because the model refinement protocol gathers information from many reconstructions instead of an individual one, but it is still our current best practice without reliable resolution estimation for continuous movement.

PDB validation metrics

In our refinement method, only metrics used in the PDB validation server are implemented to constrain the geometry of the molecules. Since many of the PDB validation metrics are based on statistics from decades-old literature with small sample sizes[24,38,44], it is debatable whether achieving a high validation score is necessary or sufficient for a good protein/RNA structure. For most of the validation metrics, the allowed std is much smaller than the resolution of even the best structures determined using CryoEM. For example, the std of most bond lengths is around 0.02Å, so even an outlier would be only ~0.1Å off from a structure with the ideal geometry. Very

often, the difference between models with a good and poor score can be so tiny that they have virtually the same similarity score when compared to a CryoEM map at near-atomic resolution. While our method can produce molecular models with near-perfect validation scores, the actual quality of the models still depends on how accurate the stereochemical statistics are, and the model we generate can be biased if the validation metrics we use are not reliable.

To show the sensitivity of the validation metrics, in a more extreme example, we started from a model of apoferritin (PDB-8T4Q) with a perfect PDB validation score and refined it with an inverted geometry loss function. That is, the optimizer will search locally, starting from a good model, and look for a model with the worst possible geometry score. Surprisingly, a model with an extremely poor score at every metric can be produced, which has only 0.2Å RMSD on average from the input good model, and no atoms shift more than 0.3Å (Figure S7). With such small movement, the two models would be virtually indistinguishable even when the map is determined at near-atomic resolution. As a result, for most CryoEM structures, the geometry scores of the molecular models are actually driven by the set of validation metrics, as well as the model refinement method used, instead of the actual information from the experimental maps.

Additionally, since PDB validation currently does not include scores for the geometry of DNA (other than bond length and angle) or many small molecule ligands, the refinement of those features is only driven by the CryoEM maps. While improving the validation metrics will be an ongoing effort of the broad structure biology community, the DNN implementation of our method makes it relatively convenient to incorporate new, well-defined metrics as a part of the loss function during training, so the quality of output models can improve as the field moves forward.

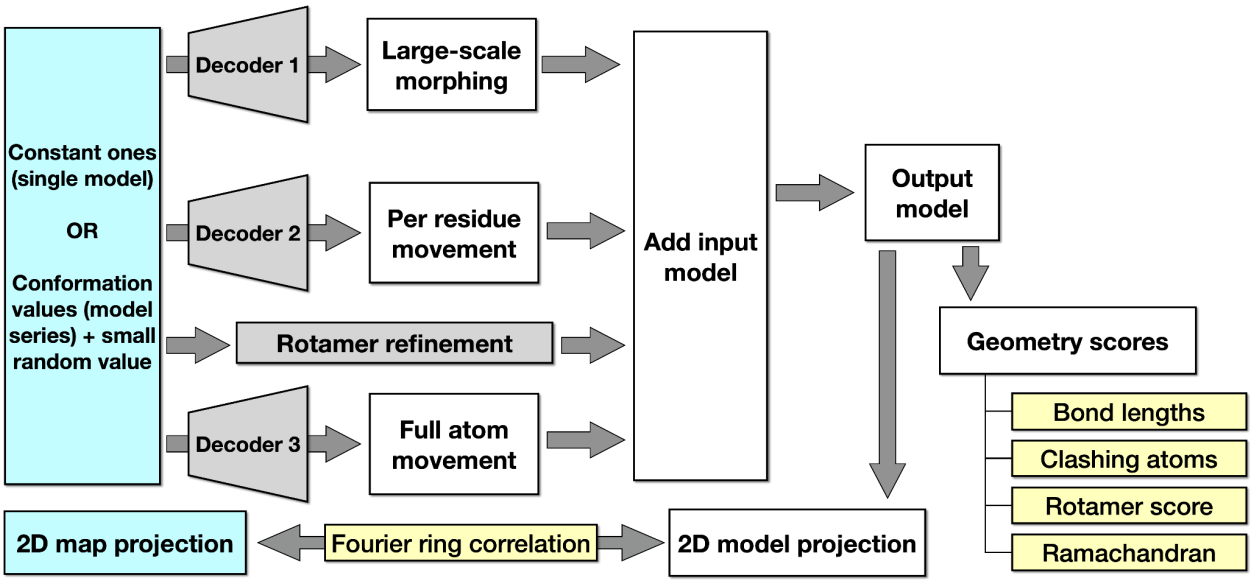

Figure S1. Workflow diagram for molecular model refinement protocol.

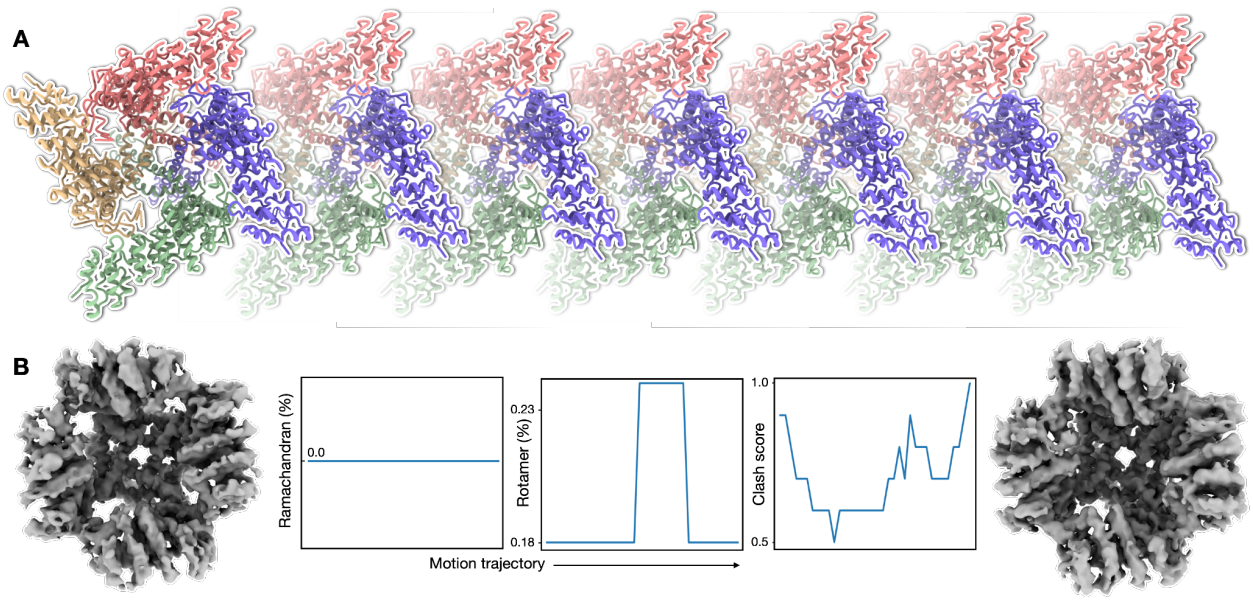

Figure S2. Model series refinement of TRPV1. **(A)** Snapshots of GMM refined molecular model series along one continuous motion trajectory of TRPV1 from dataset EMPIAR-10059. **(B)** Left and right panels - 3D reconstruction of the first and last frame of the motion; middle three panels - Ramachandran, rotamer outlier, and clash score for models along the motion trajectory.

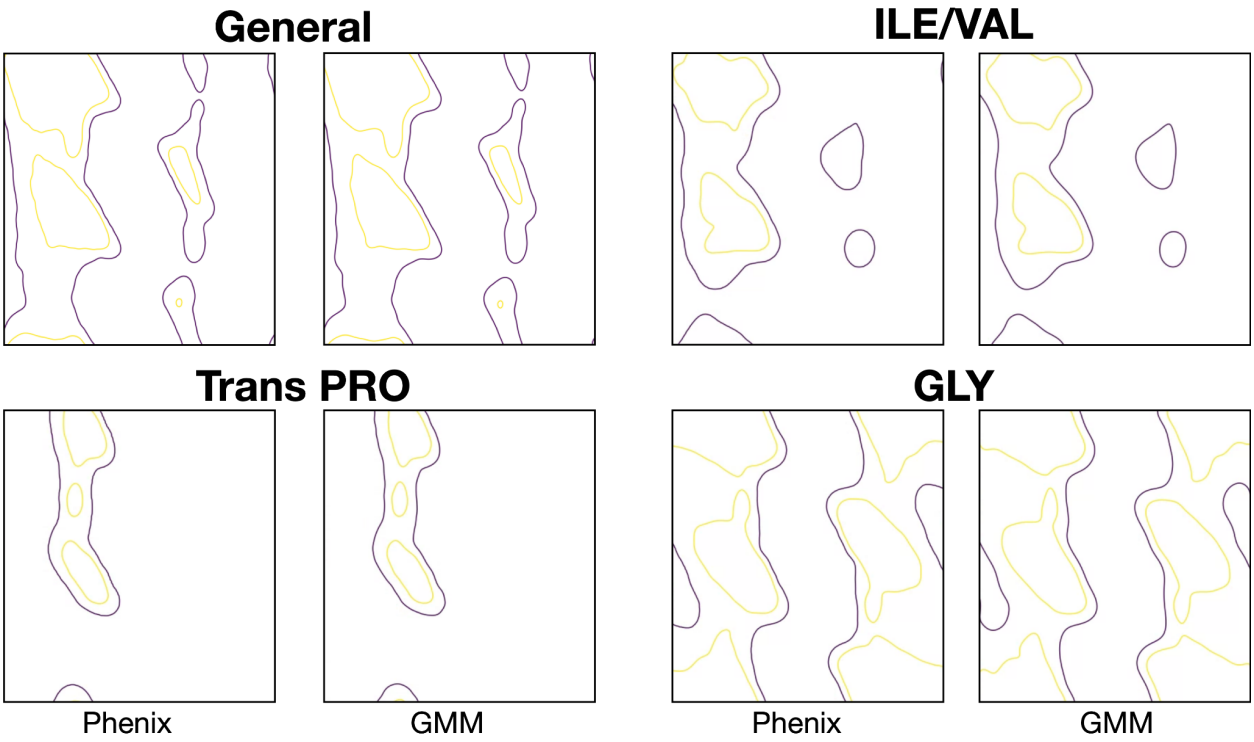

Figure S3. Comparison between histogram representation of the Ramachandran plot from Phenix, and the GMM representation of the corresponding plot. Four types of Ramachandran plots are shown, and the contour lines indicate the boundary of allowed (yellow, 2%) and outlier (purple, 0.05%). The Phenix and GMM plots of the same type should be nearly identical.

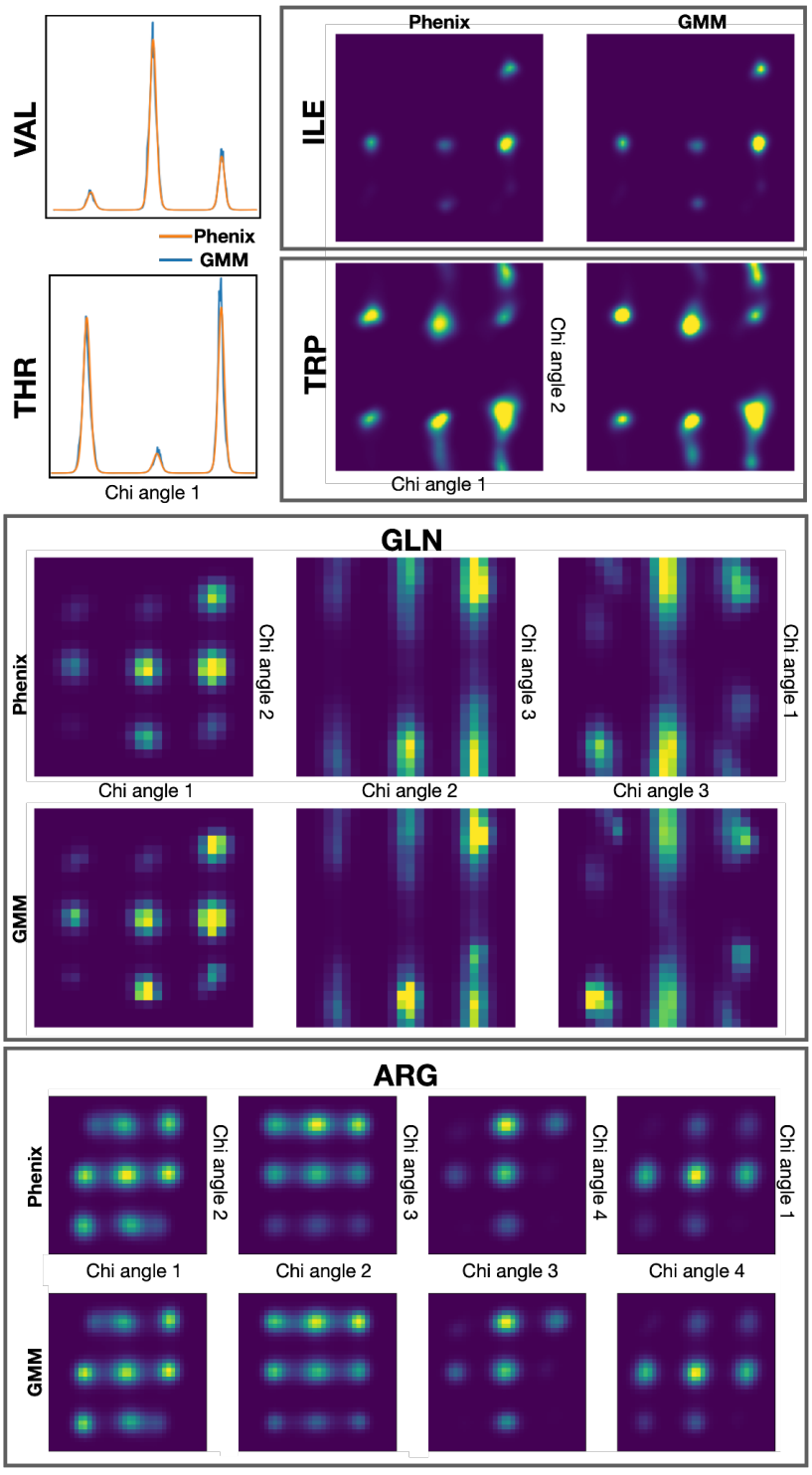

Figure S4. Comparison between histogram representation of the preferred rotamer from Phenix, and the GMM representation of the corresponding plot. Here we show two types of sidechains with one Chi angle (VAL, THR), two types with two Chi angles (ILE, TRP), one type with three Chi angles (GLN), and one with four Chi angles (ARG).

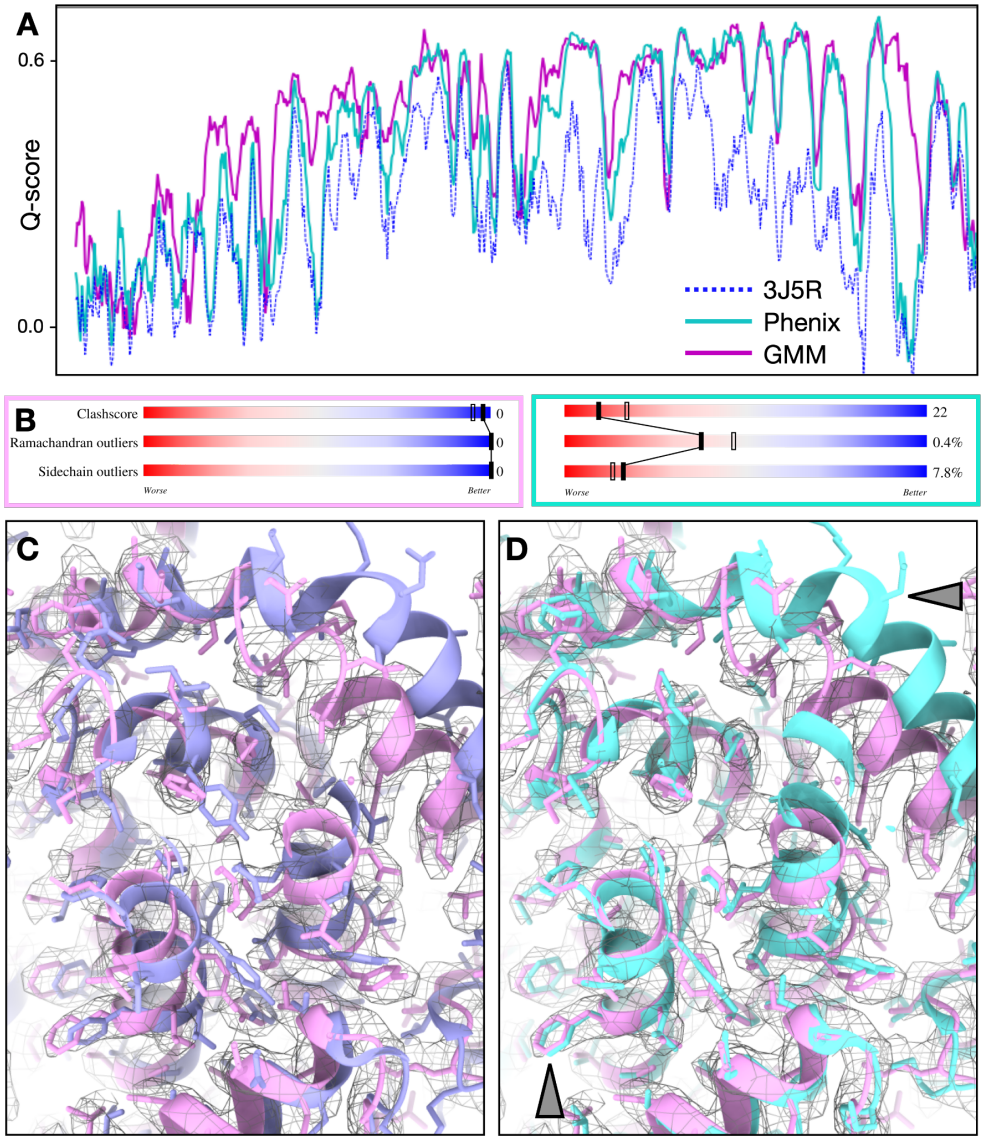

Figure S5. Comparison between the GMM-based model refinement and Phenix real-space refinement in the TRPV1 example shown in Figure 1. **(A)** Q-score comparison. Blue - original PDB model; cyan - Phenix real space refinement; pink - GMM-based refinement. **(B)** PDB validation metric comparison. Left - GMM-based refinement; right - Phenix real-space refinement. **(C)** Overlay of the original (blue) and GMM refined (pink) model with the CryoEM map. **(D)** Overlay of the original (blue) and Phenix refined (cyan) model with the CryoEM map.

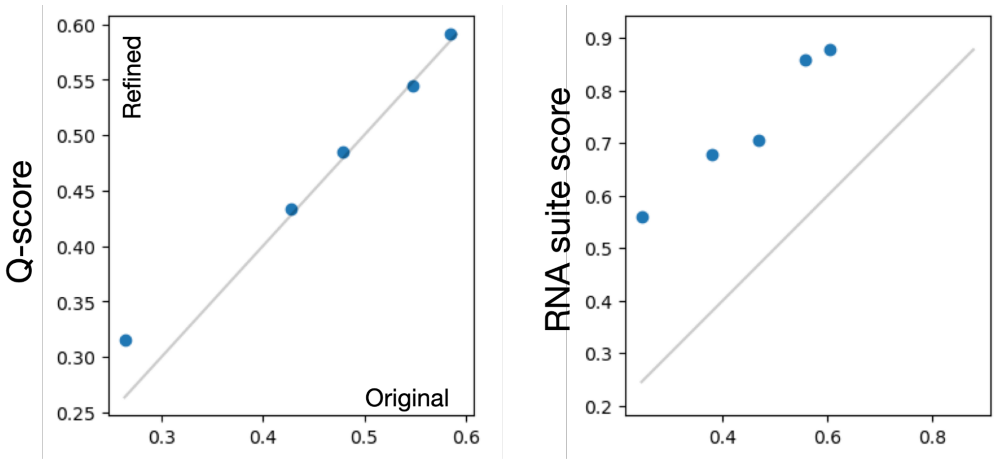

Figure S6. Comparison of the original and refined version of 5 models with RNA from the PDB. Left - Q-score; right - RNA suite score (higher score indicates better RNA backbone geometry).

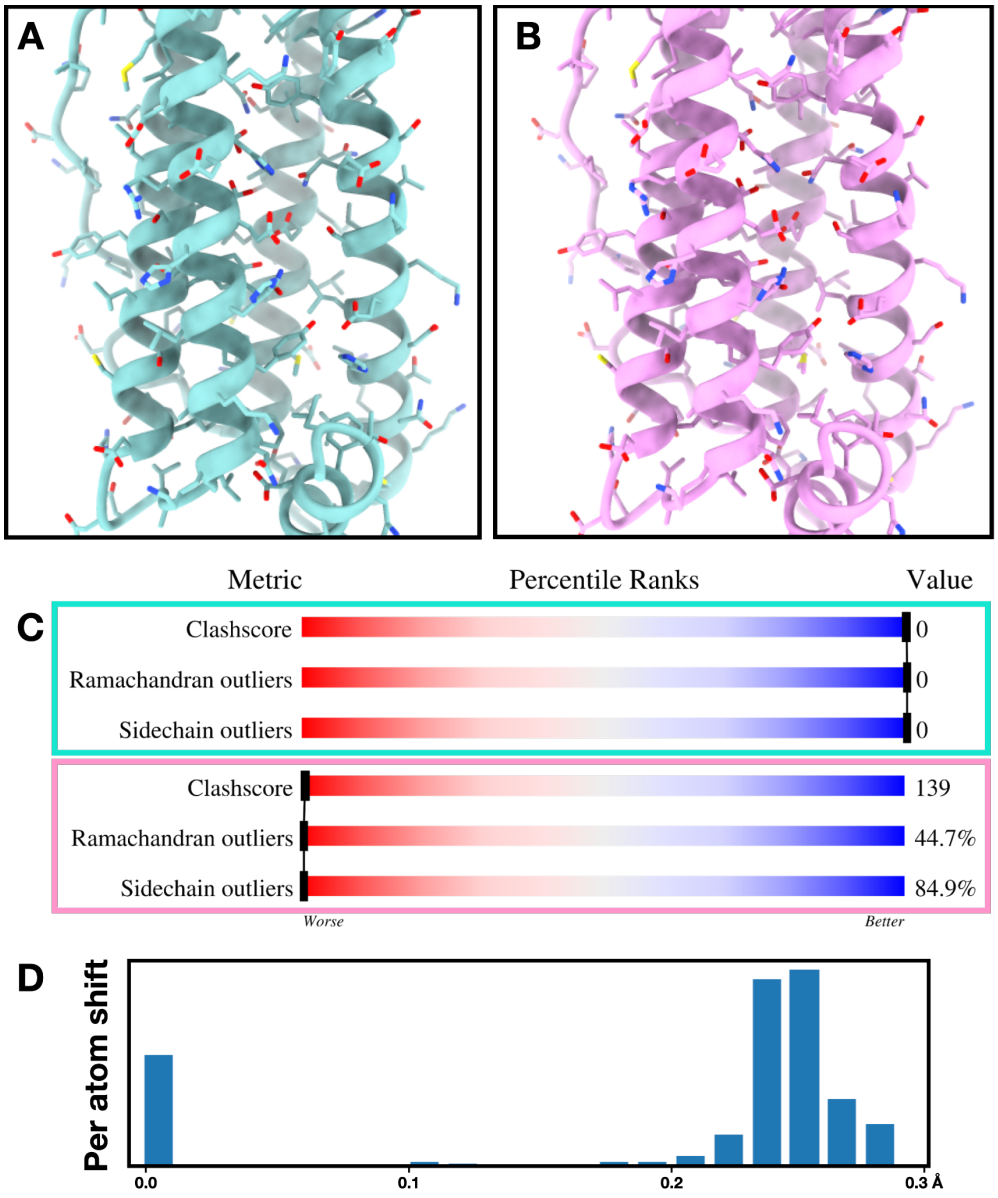

Figure S7. Sensitivity of the PDB validation metrics. (A) Input model of apoferritin (PDB-8T4Q). (B) Output model from the GMM-based refinement with an inverse loss function, using the model from A as input. (C) Comparison of PDB validation scores between the input (top) and inverse "refined" model (bottom). (D) Histogram of per-atom shift distance between the two models.

Supplementary video 1. Model series generated from the TRPV1 dataset (EMPIAR-10059), along a linear trajectory in the conformational space.

Supplementary video 2. Model series generated from the spliceosome dataset (EMPIAR-10180), along a linear trajectory in the conformational space.

Supplementary video 3. Model series generated from the spliceosome dataset (EMPIAR-10180), along a circular trajectory in the conformational space.

References

1. Cheng, Y. Single-particle cryo-EM-How did it get here and where will it go. *Science* **361**, 876–880 (2018).
2. Nakane, T. *et al.* Single-particle cryo-EM at atomic resolution. *Nature* **587**, 152–156 (2020).
3. Su, Z. *et al.* Cryo-EM structures of full-length Tetrahymena ribozyme at 3.1 Å resolution. *Nature* **596**, 603–607 (2021).
4. Davis, J. H. *et al.* Modular Assembly of the Bacterial Large Ribosomal Subunit. *Cell* **167**, 1610-1622.e15 (2016).
5. Qin, B. *et al.* Cryo-EM captures early ribosome assembly in action. *Nat. Commun.* **14**, 898 (2023).
6. des Georges, A. *et al.* Structural Basis for Gating and Activation of RyR1. *Cell* **167**, 145-157.e17 (2016).
7. Fan, G. *et al.* Conformational motions and ligand-binding underlying gating and regulation in IP3R channel. *Nat. Commun.* **13**, 6942 (2022).
8. Dashti, A. *et al.* Trajectories of the ribosome as a Brownian nanomachine. *Proc. Natl. Acad. Sci. U. S. A.* **111**, 17492–17497 (2014).
9. Zhong, E. D., Bepler, T., Berger, B. & Davis, J. H. CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks. *Nat. Methods* **18**, 176–185 (2021).
10. Chen, M. & Ludtke, S. J. Deep learning-based mixed-dimensional Gaussian mixture model for characterizing variability in cryo-EM. *Nat. Methods* **18**, 930–936 (2021).
11. Punjani, A. & Fleet, D. J. 3DFlex: determining structure and motion of flexible proteins from cryo-EM. *Nat. Methods* (2023) doi:10.1038/s41592-023-01853-8.
12. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).

13. Waterhouse, A. *et al.* SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Res.* **46**, W296–W303 (2018).
14. Jamali, K. *et al.* Automated model building and protein identification in cryo-EM maps. *Nature* **628**, 450–457 (2024).
15. Terashi, G., Wang, X., Prasad, D., Nakamura, T. & Kihara, D. DeepMainmast: integrated protocol of protein structure modeling for cryo-EM with deep learning and structure prediction. *Nat. Methods* **21**, 122–131 (2024).
16. Afonine, P. V. *et al.* Real-space refinement in PHENIX for cryo-EM and crystallography. *Acta Crystallogr D Struct Biol* **74**, 531–544 (2018).
17. Croll, T. I. ISOLDE: a physically realistic environment for model building into low-resolution electron-density maps. *Acta Crystallogr. D Struct. Biol.* **74**, 519–530 (2018).
18. Emsley, P., Lohkamp, B., Scott, W. G. & Cowtan, K. Features and development of Coot. *Acta Crystallogr. D Biol. Crystallogr.* **66**, 486–501 (2010).
19. Gore, S. *et al.* Validation of structures in the protein data bank. *Structure* **25**, 1916–1927 (2017).
20. Murphy, B. J. *et al.* Rotary substates of mitochondrial ATP synthase reveal the basis of flexible F1-Fo coupling. *Science* **364**, eaaw9128 (2019).
21. Bai, X.-C., Yan, Z., Wu, J., Li, Z. & Yan, N. The Central domain of RyR1 is the transducer for long-range allosteric gating of channel opening. *Cell Res.* **26**, 995–1006 (2016).
22. Chen, M., Toader, B. & Lederman, R. Integrating Molecular Models Into CryoEM Heterogeneity Analysis Using Scalable High-resolution Deep Gaussian Mixture Models. *J. Mol. Biol.* **435**, 168014 (2023).
23. Fabiola, F. & Chapman, M. S. Fitting of high-resolution structures into electron microscopy reconstruction images. *Structure* **13**, 389–400 (2005).
24. Chen, V. B. *et al.* MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Crystallogr. D Biol. Crystallogr.* **66**, 12–21 (2010).

25. Cao, E., Liao, M., Cheng, Y. & Julius, D. TRPV1 structures in distinct conformations reveal activation mechanisms. *Nature* **504**, 113–118 (2013).
26. Gao, Y., Cao, E., Julius, D. & Cheng, Y. TRPV1 structures in nanodiscs reveal mechanisms of ligand and lipid action. *Nature* **534**, 347–351 (2016).
27. Pintilie, G. *et al.* Measurement of atom resolvability in cryo-EM maps with Q-scores. *Nat. Methods* **17**, 328–334 (2020).
28. Plaschka, C., Lin, P.-C. & Nagai, K. Structure of a pre-catalytic spliceosome. *Nature* **546**, 617–621 (2017).
29. Nakane, T., Kimanius, D., Lindahl, E. & Scheres, S. H. Characterisation of molecular motions in cryo-EM single-particle data by multi-body refinement in RELION. *Elife* **7**, (2018).
30. Punjani, A. & Fleet, D. J. 3D variability analysis: Resolving continuous flexibility and discrete heterogeneity from single particle cryo-EM. *J. Struct. Biol.* **213**, 107702 (2021).
31. Roh, S.-H. *et al.* Cryo-EM and MD infer water-mediated proton transport and autoinhibition mechanisms of Vo complex. *Sci. Adv.* **6**, eabb9605 (2020).
32. Zhang, K. *et al.* Structure of the 30 kDa HIV-1 RNA Dimerization Signal by a Hybrid Cryo-EM, NMR, and Molecular Dynamics Approach. *Structure* (2018) doi:10.1016/j.str.2018.01.001.
33. Gauto, D. F. *et al.* Integrated NMR and cryo-EM atomic-resolution structure determination of a half-megadalton enzyme complex. *Nat. Commun.* **10**, 2697 (2019).
34. Chen, M., Schmid, M. F. & Chiu, W. Improving resolution and resolvability of single-particle cryoEM structures using Gaussian mixture models. *Nat. Methods* (2023) doi:10.1038/s41592-023-02082-9.
35. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv [cs.LG]* (2014).
36. Adams, P. D. *et al.* PHENIX: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr. D Biol. Crystallogr.* **66**, 213–221 (2010).

37. Yamashita, K., Palmer, C. M., Burnley, T. & Murshudov, G. N. Cryo-EM single-particle structure refinement and map calculation using Servalcat. *Acta Crystallogr. D Struct. Biol.* **77**, 1282–1291 (2021).
38. Richardson, J. S. *et al.* RNA backbone: consensus all-angle conformers and modular string nomenclature (an RNA Ontology Consortium contribution). *RNA* **14**, 465–481 (2008).
39. Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. (2013).
40. Goddard, T. D. *et al.* UCSF ChimeraX: Meeting modern challenges in visualization and analysis. *Protein Sci.* **27**, 14–25 (2018).
41. wwPDB consortium. Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Res.* **47**, D520–D528 (2019).
42. Phillips, J. C. *et al.* Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **26**, 1781–1802 (2005).
43. Pettersen, E. F. *et al.* UCSF Chimera--a visualization system for exploratory research and analysis. *J. Comput. Chem.* **25**, 1605–1612 (2004).
44. Word, J. M. *et al.* Visualizing and quantifying molecular goodness-of-fit: small-probe contact dots with explicit hydrogen atoms. *J. Mol. Biol.* **285**, 1711–1733 (1999).