Annotating precision for integrative structural models using deep learning

Nikhil Kasukurthi[1] and Shruthi Viswanath[1*]

[1]National Center for Biological Sciences, Tata Institute of Fundamental Research, Bangalore, India

*To whom correspondence should be addressed.

Short title: Precision for integrative models

Keywords: model precision, integrative structure modeling, validation of integrative models, detecting substructures, deep learning, autoencoders.

Abstract

Motivation: Integrative modeling of macromolecular structures usually results in an ensemble of models that satisfy the input information. The model precision, or variability among these models is estimated globally, i.e., a single precision value is reported for the model. However, it would be useful to identify regions of high and low precision. For instance, low-precision regions can suggest where the next experiments could be performed and high-precision regions can be used for further analysis, e.g., suggesting mutations.
Results: We develop PrISM (Precision for Integrative Structural Models), using autoencoders to efficiently and accurately annotate precision for integrative models. The method is benchmarked and tested on five examples of binary protein complexes and five examples of large protein assemblies. The annotated precision is shown to be consistent with, and more informative than localization densities. The generated networks are also interpreted by gradient-based attention analysis.
Availability: Source code is at https://github.com/isblab/prism.
Contact: shruthiv@ncbs.res.in

1 Introduction

Integrative modeling, also known as hybrid modeling, has emerged as the method of choice for determining the structures of large macromolecular assemblies which are challenging to characterize using a single experimental method such as X-ray crystallography or cryo-electron microscopy (Alber *et al.*, 2007; Ward *et al.*, 2013; Rout and Sali, 2019). Data from complementary experiments is combined with physical principles, statistical inference, and existing models to produce the structure (Russel *et al.*, 2012; Webb *et al.*, 2018; Rout and Sali, 2019; Saltzberg *et al.*, 2021). Several large assemblies have been recently determined by this approach, yielding insights on various processes such as transcription (Robinson *et al.*, 2015), gene regulation and DNA repair (Luo *et al.*, 2015), intra-cellular transport (Kim *et al.*, 2018; Ganesan *et al.*, 2020), cell cycle progression (Viswanath, Bonomi, *et al.*, 2017), immune response and metabolism (Lasker *et al.*, 2012; Gutierrez *et al.*, 2020).

Integrative modeling often relies on data that is sparse, noisy, ambiguous, and from heterogenous samples (Schneidman-Duhovny *et al.*, 2014). Consequently, there is usually more than one model (structure) that satisfies the data. Therefore, an important attribute of an integrative model is its precision, defined as the variability among the models that satisfies the input data. The precision defines the uncertainty of the structure and is a lower bound on its accuracy. Importantly, downstream applications of the structure are limited by its precision. For example, one cannot use a protein model of 20 Å precision to accurately identify binding sites for drug molecules. Further, precision aids in making informed choices for future modeling, including altering the representation, degrees of freedom, or the amount of sampling (Viswanath, Chemmama, *et al.*, 2017).

Currently, precision is estimated globally; i.e., a single precision value is reported for the entire integrative model. However, it would be useful to identify regions of high and low precision within the model. For instance, low-precision regions can be used to suggest where the next set of experimental data should be collected to improve the model. High-precision regions can be used for further biological analysis such as identifying binding interfaces, rationalizing known mutations, and suggesting new mutations.

Several methods have been proposed for detecting substructure similarities and determining flexible/rigid regions in molecular dynamics simulations of proteins (Wriggers and Schulten, 1997; Kedem *et al.*, 1999; Jacobs *et al.*, 2001; Pfleger *et al.*, 2013; Martínez, 2015; Cazals and Tetley, 2019). However, these approaches are not directly applicable to integrative models of macromolecular assemblies for one or more of the following reasons. First, these methods rely on the input being a set of atomic structures with known secondary structure. In contrast, integrative models possess a more general representation encoding multi-scale coarse-grained models, and can comprise of regions with unknown atomic structure (Viswanath and Sali, 2019). Second, these methods identify rigid substructures but do not quantify precision for all parts of the structure. Finally, these methods analyze structures with a single or small number of proteins and have not been demonstrated to be scalable for analyzing tens of thousands of models of large assemblies.

Our method, PrISM (Precision for Integrative Structural Models), uses an autoencoder, a type of unsupervised deep neural network, to efficiently annotate precision for regions of an integrative

model. Autoencoders were first used for denoising images in computer vision by learning a low-dimensional or compressed representation of the input images (Hinton and Salakhutdinov, 2006). Characteristics that are common to most inputs (signal) are captured in the compressed representation; in contrast, noise varies between inputs and is not captured. Therefore, autoencoders are a convenient choice of method for discriminating signal from noise in the input. For a set of integrative models, high-precision regions, whose configurations (spatial positions) are relatively unchanged across models, represent signal; in contrast, low-precision regions, whose configurations vary considerably across the models, represent noise.

Variational autoencoders have been previously used in structural biology for clustering protein structures from molecular dynamics simulations to visualize intermediates in protein folding (Doerr *et al.*, 2017; Hernández *et al.*, 2018; Bhowmik *et al.*, 2018; Chen *et al.*, 2020) identifying changes in dynamics upon binding (Tsuchiya *et al.*, 2019; Romero *et al.*, 2019; Chen *et al.*, 2020). They have also been used to accelerate molecular dynamics simulations (Lee *et al.*, 2019) and as generative models for protein structure prediction (Hoseini *et al.*, 2021; Eguchi *et al.*, 2020) . These methods are based on convolutional neural networks with contact map or distance matrix representations of input models, while PrISM is based on fully-connected networks with one-dimensional input vectors (Methods).

The rest of this paper is structured as follows. In Methods, we describe the inputs and outputs, details of the networks, datasets, performance metrics, and the implementation of attention analysis for interpreting the networks. In Results, we demonstrate the efficiency and accuracy of PrISM on datasets of protein-protein complexes and larger assemblies. We also recommend a generic architecture for a new integrative modeling system. Results from PrISM are shown to be more informative than localization probability density maps, while being consistent with them. Further, we interpret the resultant networks using gradient-based attention analysis. Finally, in Discussion, we describe the advantages, disadvantages, and applicability of PrISM, and comment on alternate design choices and future directions.

2 Methods

2.1 Inputs and outputs

The objective at hand is to identify regions of high and low precision in an ensemble of integrative structural models. Accordingly, the input for PrISM is a set of structurally superposed integrative models (Fig. 1). While integrative models can be represented by a set of atomic coordinates, frequently, a coarse-grained representation is used (Viswanath and Sali, 2019; Sali *et al.*, 2015; Vallat *et al.*, 2018). Proteins in the latter representation are represented by a sequence of spherical beads, where a bead corresponds to a number of contiguous residues along the protein sequence. Our method is applicable to both types of integrative models, atomic and coarse-grained; atomic models are internally converted to residue-level beads centered at the corresponding Cα atoms. Further, the input ensemble can consist of any set of models whose precision one is interested in analyzing. For instance, it could be the set of models from the most populated cluster of the integrative modeling analysis pipeline (Viswanath, Chemmama, *et al.*, 2017). The output from PrISM is a quantitative measure of precision per bead as well as a visualization of regions of high and low precision in the ensemble as a heat map overlaid on a representative structure (Fig. 1).

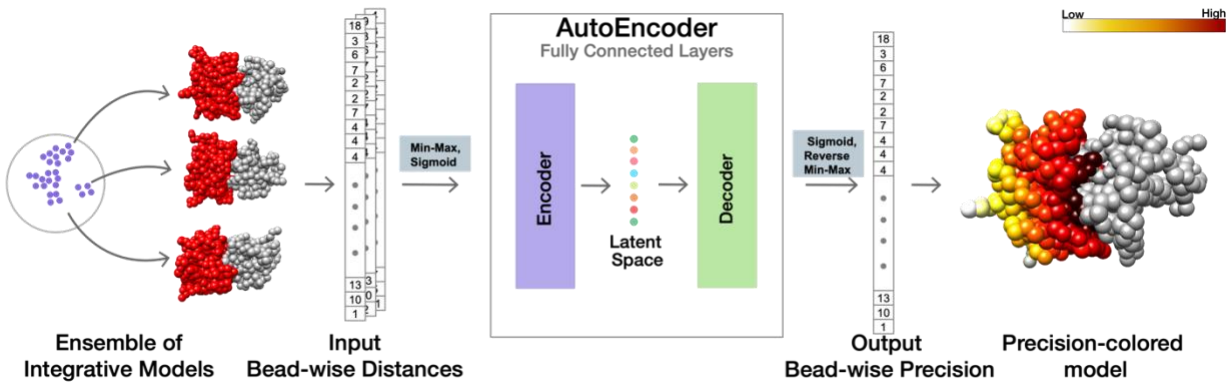

Fig. 1. Schematic of PrISM. The input is a set of structurally superposed integrative models, each converted to a one-dimensional vector of per-bead distances. These vectors are normalized and subsequently passed to the encoder, latent space, and decoder in sequence for training the autoencoder. Once training is complete, a random input vector passed to the autoencoder results in an output which, upon rescaling, produces the bead-wise precision of the ensemble, which can be visualized on a structural model.

2.2 Details of neural networks

2.2.1 Schematic

Auto-encoders are unsupervised deep neural networks that reconstruct the given input (Hinton and Salakhutdinov, 2006). First, the input is pre-processed and passed through an encoder, a neural network which learns the features that represent the input. These features are passed to the latent space or the bottleneck layer, which is a compressed, *i.e.*, lower dimension representation of the input. Finally, the output of the latent space is passed to the decoder, a neural network through which the compressed representation is reconstructed to the original input. Once the autoencoder is trained on a set of input models, a random input passed through it produces an output (reconstruction) that summarizes the input models (Fig. 1).

2.2.2 Processing the inputs and outputs

Given the set of *N* input models with *M* beads each, each model is converted to an $M \times 1$ vector, encoding the distance of each bead in the model to its centroid in the set of input models. For the *i*th bead of the *j*th model, an element of this input vector, d_{ij} is defined as

$$d_{ij} = \left\| \overrightarrow{r_{ij}} - \frac{\sum_{k=1}^{N} \overrightarrow{r_{ik}}}{N} \right\| \tag{1}$$

Each vector element, d_{ij}, is min-max normalized using the minimum and maximum distances for each bead and passed through a sigmoid function to obtain a smooth output between 0 and 1 before passing to the encoder. Similarly, at the other end, the output from the decoder is passed through a sigmoid function and reverse min-max rescaled per bead to obtain the reconstructed output in the same scale as the input.

2.2.3 Types of networks

We trained three types of autoencoders: autoencoders with the default loss function of mean-squared error (AE) (Hinton and Salakhutdinov, 2006), autoencoders with the MMD loss (AE-

MMD) (Zhao *et al.*, 2018), and variational autoencoders (VAE) (Kingma and Welling, 2014). The simplest autoencoders are trained on the mean-squared error (MSE) loss which is based on the Euclidean distance between the reconstructed input and the original input. AE-MMDs additionally have a maximum mean discrepancy (MMD) term in the loss (Zhao *et al.*, 2018). The MMD quantifies the difference between the input distribution and the latent space compressed input distribution in terms of the difference in their moments (Zhao *et al.*, 2018). Adding the MMD is shown to result in a more informative latent space with reduced variance (Zhao *et al.*, 2018). Finally, VAEs are autoencoders where the latent space is an approximation of the normal distribution. In addition to the MSE, the VAE loss function has a regularization term based on the Kullback Leibler divergence between the latent space compressed input distribution and the normal distribution (Kingma and Welling, 2014).

2.2.4 Hyperparameters

The encoder and decoder are fully-connected networks with multiple layers each. The first layer of the encoder has 32 neurons; each subsequent layer adds neurons in powers of 2 (i.e., 32, 64, 128, 256, etc.). The decoder has an identical architecture. The number of encoder and decoder layers and the number of neurons in the latent space are the three varied hyperparameters. The choices for the number of encoder and decoder layers were 2, 3, 4, and 5. The number of latent space neurons was varied in powers of 2 starting from 8 (8, 16, 32, 64, 128, 256). A grid search was performed to identify the optimal hyperparameters for each system and type of network (AE, AE-MMD, VAE), varying one hyperparameter at a time (Supplementary Table S1). The architecture with the best accuracy (Evaluation metrics, below) was chosen for each system and type of network. All the networks were trained with a constant learning rate of 1e-3 using Adam optimizer (Kingma and Ba, 2017) and batch size of 64 for 50 epochs on an Nvidia Quatro P4000 (8 GB VRAM).

2.3 Datasets

First, the search for hyperparameters, evaluation, and benchmarking was performed on five binary protein complexes and three large protein assemblies. The binary complexes were earlier used to establish integrative modeling protocols and included: complex between trypsin and its inhibitor (PDB 1AVX), RAN and RCC1 (PDB 1I2M), colicin and its inhibitor (PDB 7CEI), spliceosomal protein and its inhibitor (PDB 1SYX), and subunits of DNA polymerase III (PDB 2IDO) (Viswanath, Chemmama, *et al.*, 2017; Viswanath and Sali, 2019). The three large protein assemblies included: complex between actin, gelsolin and tropomysin (Saltzberg *et al.*, 2019), yeast γ-tubulin small complex bound to Spc110 (Brilot *et al.*, 2021), and the transcription and DNA repair factor TFIIH (Luo *et al.*, 2015). PrISM was further evaluated on an independent dataset of two large assemblies including models of RNA polymerase II from two different model analysis pipelines, an older one implementing k-means clustering (Webb *et al.*, 2018) and another more recent one implementing distance threshold-based clustering (Saltzberg *et al.*, 2021). For each system, the input was the set of models from the most populated cluster from IMP analysis (Viswanath, Chemmama, *et al.*, 2017; Saltzberg *et al.*, 2021). All these datasets are available in public repositories on https://integrativemodeling.org

2.4 Evaluation metrics

The accuracy of the precision annotated by PrISM was evaluated using the R^2 score (Heinisch, 1962) and Matthew's correlation co-efficient (Matthews, 1975). The bead-wise precision output

by PrISM approximates the standard deviation of the distance of each bead to its centroid in the model ensemble; henceforth this standard deviation is termed the spread, denoted by s. High- (low-) precision beads have low (high) spread. Therefore, we compared the AE output to the spread. The spread was first transformed to make it comparable to the AE output by applying a sigmoid function on it and subsequently rescaling the sigmoid output to the range of the original bead distance. We computed the R^2 score between the AE output and the spread, thus measuring the correlation between them. Additionally, to assess the quality of the annotation in a coarser way, we classified beads as high and low precision and assessed if PrISM could correctly predict the high and low precision beads using Matthew's correlation coefficient, a measure of accuracy for binary classification (Matthews, 1975). To obtain the reference high and low precision beads, we computed the mean μ and standard deviation σ of the distribution of the spread s across all beads. We then defined high precision beads as those for which the spread was lower than one standard deviation from the mean, *i.e.*, $s \leq \mu(s) - \sigma(s)$ and low-precision beads as those for which the spread was higher than one standard deviation from the mean, *i.e.*, $s \geq \mu(s) + \sigma(s)$. Further, to assess the efficiency of PrISM, the number of model parameters and training times were computed. The precision annotation was also compared to localization probability density maps (Viswanath, Chemmama, *et al.*, 2017; Saltzberg *et al.*, 2021).

2.5 Attention analysis

Neural networks are considered as black-boxes since they cannot be easily interpreted. To interpret the networks in PrISM, we adopted the method of gradient-based attention used to interpret VAEs designed for image recognition (Liu *et al.*, 2020). The gradients of each latent space neuron (or latent dimension) are backpropagated to the last fully-connected layer in the encoder to obtain an attention map corresponding to an individual latent dimension; additionally, the overall attention map is the mean of the maps for each latent dimension. The attention map is a way to visualize the relative weight (attention) given to different parts of the input during training.

3 Results

We first demonstrate the performance of PrISM on benchmarks of five binary complexes and three large assemblies. Then, we recommend a generic architecture for a new system and evaluate PrISM on an independent test set of two assemblies. Next, we compare the annotated precision with localization probability density maps. Finally, we interpret the networks using gradient-based attention analysis.

3.1 Performance evaluation on benchmarks

For each type of network (AE, AE-MMD, VAE) and each biological system, we tuned the hyper-parameters (Supplementary Table S1, Methods). We then compared the performance of the networks (Fig. 2). Overall, all the examined networks are accurate on binary complexes (R^2 and Matthew's correlation co-efficient both higher than 0.9) (Figs. 2A, 2B), while the training time is lower for autoencoders (AE) (Fig. 2C). The performance on larger assemblies shows that the method is scalable and precision can be annotated in a few minutes for such assemblies on a modern workstation (Figs. 2D, 2E, 2F). The different networks are on average equally accurate on actin and Spc110-γTuSC, while the R^2 is slightly lower for autoencoders on TFIIH but still higher than 0.85 (Figs. 2D, 2E). Consistent with the binary complexes, the training time is lower for autoencoders (AE) (Fig. 2F). Overall, autoencoders have an accuracy comparable to the other

networks, while demonstrating consistently low training times across datasets. Therefore, we chose autoencoders for annotating precision for a new system.

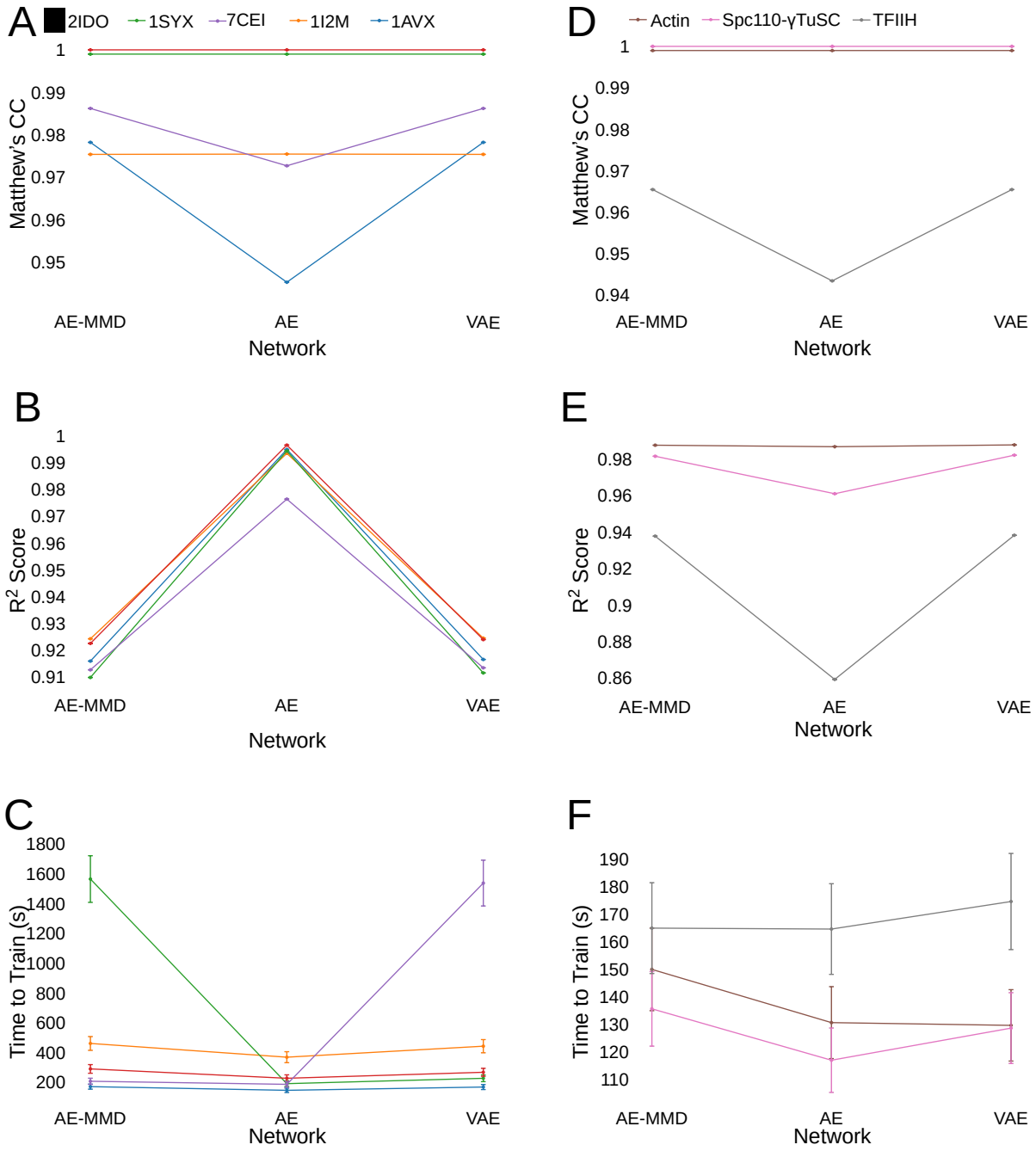

Fig. 2. Results on binary protein complexes and large assemblies. Performance is assessed on five binary protein complexes (2IDO, 1SYX, 7CEI, 1I2M, 1AVX) and three large complexes (Actin, Spc110-γTuSC, TFIIH) for each type of network (AE-MMD, AE, VAE) after hyperparameter tuning. Accuracy is assessed in terms of (A, D) Matthew's correlation co-efficient, and (B, E) R^2 score. The efficiency of the network is measured in terms of training time (C, F) . Higher values are better for panels (A, B, D, E); lower values are better for panels (C, F). All the networks were trained with a constant learning rate of 1e-3, batch size of 64 for 50 epochs on an Nvidia Quatro P4000 (8 GB VRAM). Networks were trained thrice to compute the error bars.

3.2 Recommended architecture and performance on independent test set

An extensive hyperparameter search was performed for each complex (Supplementary Table 1), based on which the architecture of the best performing networks was obtained (Table 1). Based on the benchmarks, 2 encoder layers and 3 decoder layers seem to be sufficient for most systems. On average, 32 latent space neurons are required for small complexes of 2-3 proteins, while 128 neurons maybe recommended for large assemblies.

Table 1. Architecture of the best-performing autoencoders The architecture of the best-performing autoencoder for each system determined by hyperparameter search is shown.

Complex	Encoder layers	Latent space neurons	Decoder layers
1AVX	2	32	2
1I2M	3	128	4
1SYX	2	16	3
2IDO	5	8	2
7CEI	2	32	2
Actin	2	8	4
Spc110-γTuSC	2	128	2
TFIIH	2	128	3

Using an autoencoder (AE) with the recommended architecture of 2 encoder layers, 3 decoder layers and 64 latent space neurons, we evaluated PrISM on a separate test set of two assemblies. The corresponding R^2 score, Matthew's correlation co-efficient and training times were 0.93, 0.83, 55s for the RNA Polymerase II ensemble in (Webb *et al.*, 2018) and 0.89, 0.85, 375s for the RNA polymerase II ensemble in (Saltzberg *et al.*, 2021).

3.3 Comparison to localization density maps

The precision-colored bead models from PrISM were compared to localization probability density maps which provide an alternative visualization of the ensemble of input models (Fig. 3). These density maps specify the probability of any volume element being occupied by a given bead in superposed models. In the first example, the input ensemble comprises of integrative models of the binary complex, 1I2M, obtained by docking a rigidly mobile ligand protein (RAN) on a fixed receptor protein (RCC1) based on chemical crosslinks, excluded volume, and sequence connectivity (Fig. 3A). The localization density map shows various configurations for RAN on RCC1. It indicates that the interface residues of RAN involved in binding to RCC1 are approximately consistent across models, while the binding site on RCC1 varies. The precision-colored model captures the same information; beads representing interface residues are of high precision (dark-colored RAN beads closest to RCC1), while beads which are further away from the interface are more mobile and of low precision (light-colored RAN beads away from RCC1).

The second example demonstrates the method on integrative models obtained by docking a monomer of Spc110-N terminus to the fixed four-protein γ-tubulin small complex using chemical crosslinks, excluded volume, sequence connectivity, and distance restraints. The localization density maps of Spc110-N terminus and the corresponding precision-colored model are compared

(Fig. 3B). The helix of Spc110 is kept fixed during simulation, indicated by a precise localization in the density maps (green helix and narrow green density at the bottom center of densities, Fig. 3B). This is consistent with the PrISM output where the helix is shown as a high-precision region (black beads, bottom center of precision-colored model, Fig. 3B). On the other hand, N-terminal to the helix is a region of unknown structure predicted to be disordered and represented by flexible beads. This region has a broad localization in the density map indicated by the green cloud above the Spc110 helix. However, the corresponding region in the precision-colored model shows beads of several colors; for example, beads in red have a higher precision than beads in orange, yellow, and white. This indicates that at least some regions in the disordered N-terminus of Spc110 can be precisely localized on γ-TuSC based on the data. Interestingly, these precise regions correspond to the conserved centrosomin domain of Spc110 visualized in newer cryo-EM maps (Brilot *et al.*, 2021). This information, while not readily available from the localization maps, is available from the PrISM output as it is a more fine-grained visualization of the ensemble.

3.4 Interpreting the networks using attention analysis

We attempted to interpret the autoencoder networks using gradient-based attention analysis to determine what aspects of the input are emphasized during training. The average attention map across all latent space dimensions was computed and visualized for binary complex 1AVX (Fig. 4). It is observed that low-precision beads, on average, have higher attention (indicated by larger, yellower circles) than high-precision beads (indicated by smaller, bluer circles). Low-precision beads signify noise in the data, making these inputs "harder" to learn, thereby requiring additional attention during training. In other words, the training effort is focused on features distinguishing input models rather than on the common features.

The attention map was also similarly visualized for individual latent space dimensions separately (Fig. 5). For example, the 10th latent space dimension for the binary complex 1AVX focuses attention on beads 110-140, while the 27th dimension focuses attention on beads 30-40 and 90-100. This indicates that latent spaces are disentangled, with different dimensions focusing attention on different aspects of the input.

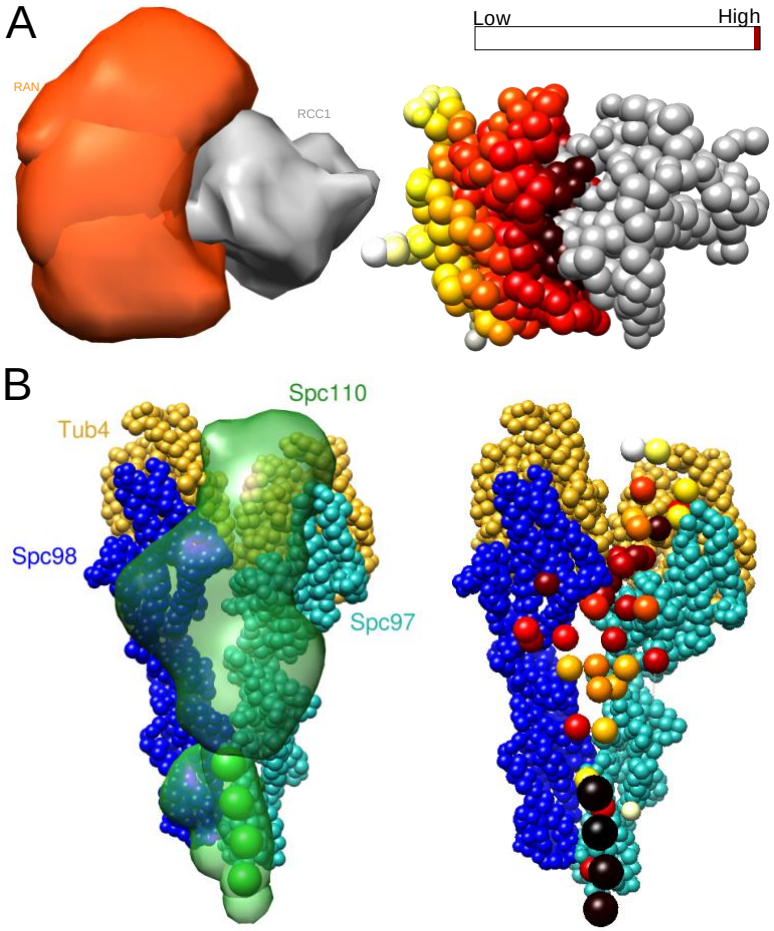

Fig. 3. Comparison to localization density maps. Two examples, (A) 1I2M, (B) Spc110- γTuSC complex, comparing localization density maps of the input ensemble with the output from PrISM visualized using precision-colored heat maps on representative bead models. Component proteins are shown in different colors.

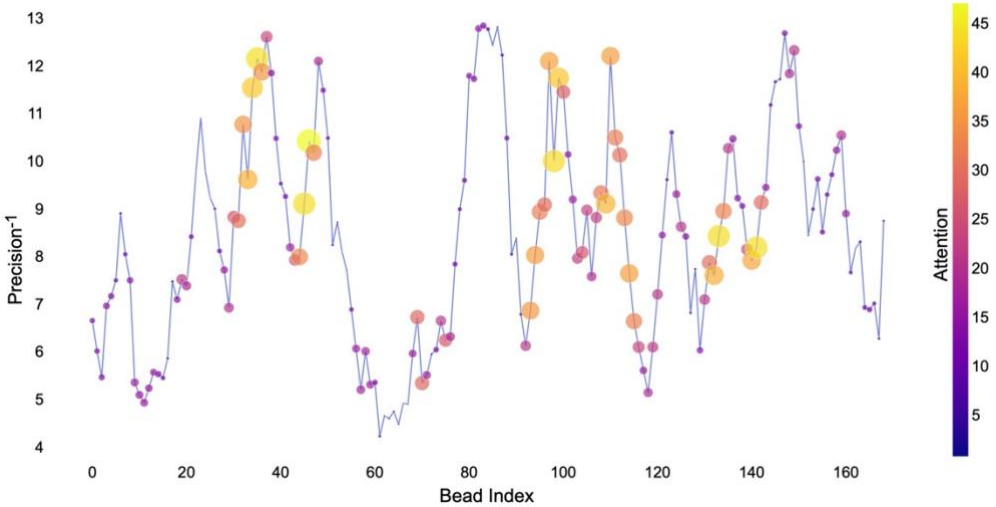

Fig. 4. Gradient-based attention correlates with precision. Plot of bead-wise inverse precision (Y-axis) overlaid with overall bead-wise attention (colors and sizes of overlaid circles) for binary complex 1AVX. Larger and yellower circles indicate higher attention. Low-precision beads are given higher attention during training.

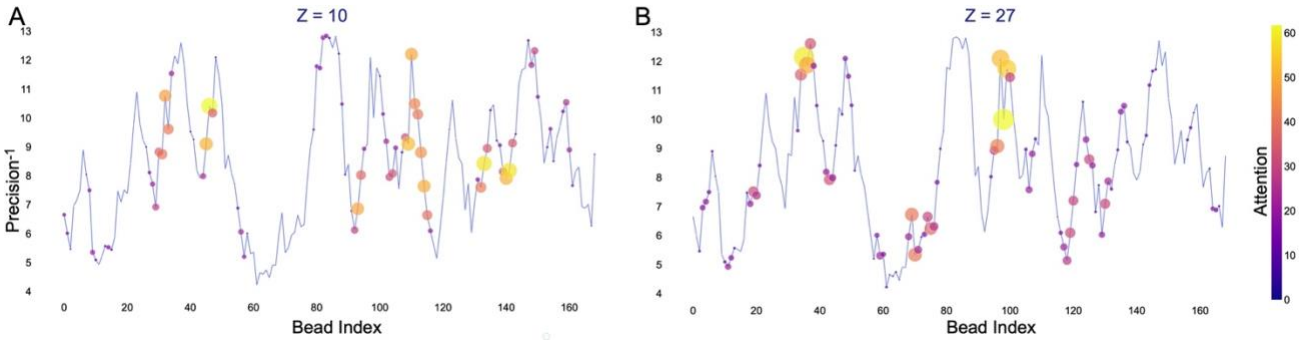

Fig. 5. Latent space disentanglement. Plot of bead-wise inverse precision overlaid with bead-wise attention for two latent space dimensions for binary complex 1AVX. The visualization of attention is similar to Fig. 5. Different dimensions in the latent space focus attention on different input regions.

4 Discussion

This work presents the first application of deep learning in integrative modeling. Our method, PrISM takes as input an ensemble of integrative models and visualizes regions of high and low precision (Fig. 1). We evaluate the method on five datasets of protein-protein complexes and three datasets of larger protein assemblies. We demonstrate that the annotation of precision is fast even for ensembles of thousands of models of large macromolecular assemblies, due to the use of optimized GPU code. The distribution of bead positions in the ensemble is accurately modeled and the results are consistent with localization probability density maps, while providing more fine-grained information. Interpretation of the resulting deep neural networks illustrates that low-precision regions receive more attention during training and different neurons in the bottleneck layer focus on different regions of the input. Autoencoders form a convenient choice of method for discriminating between high precision (signal) and low precision (noise) regions in the input.

The annotated precision provides an estimate of the uncertainty of different regions. It is useful for assessing the quality of input data, suggesting where the next set of data should be gathered, performing biological analysis such as identify binding sites or suggesting mutations, and for making informed choices of representation, degrees of freedom, and amount of sampling for future modeling (Viswanath, Chemmama, *et al.*, 2017). While presented as a tool for integrative structural models, PrISM is broadly applicable to any set of structures including atomic structures and can be used, for example, to determine flexible/rigid sub-regions.

The disadvantages of PrISM are that, first, we need to train a new network for a new system, but we show that this can be done fast, in a few minutes on a modern workstation for a large assembly. Second, the recommended values of hyperparameters are based on the examined benchmarks, which are expected to perform reasonably but are not guaranteed to be the best-performing parameters for a given system.

Alternate choices of inputs and network architectures were examined. Particularly, we explored the use of 2D contact maps or pairwise bead distance matrices as inputs. However, these types of inputs confound the precision calculation if, for example, subunits move in concert (*e.g.*, as a single rigid body) during the simulation. The difference in contact maps between different models can be miniscule for bead pairs in such subunits, even if the rigid movement is quite large in the

simulation. As a consequence, beads in such regions might be annotated as high precision while they are in fact, in low precision regions. To eliminate this inter-bead dependence on precision calculation, we used the positional variation of single beads as input. Further, integrative models can be multi-scale models where the same parts of the system are represented by beads at multiple resolutions simultaneously. We ran PrISM with input beads at different resolutions for such models, and found the output to be consistent between them.

We also explored the use of convolutional networks in place of fully-connected networks. While both networks were equally accurate, gradient-based attention analysis showed that all the latent dimensions in convolutional networks possessed the same weight, indicating redundancy (overfitting), leading us to re-examine the network architecture. The use of kernels in convolutional networks implicitly assumes that the position of a bead is dependent on the positions of beads adjacent to it in the protein sequence. However, a bead can form contacts in the structure with beads that are far away in sequence. Fully-connected networks are able to capture this characteristic since the position of a bead can be influenced by the position of any other bead. Therefore, they were found to be a better choice for the input at hand.

Validation of integrative models and data is an open research challenge. This is timely due to the new worldwide Protein Data Bank archive for integrative structures (Sali *et al.*, 2015) (http://pdb-dev.wwpdb.org). We suggest that methods such as PrISM be used in the pipeline for visualizing and assessing the precision of integrative models in the wwPDB archive. This will ultimately improve the quality and utility of deposited integrative structures.

Acknowledgements

We thank Serena Chen for introducing us to autoencoders and for useful comments on the manuscript.

Funding

This work has been supported by the Department of Atomic Energy (DAE) TIFR grant RTI 4006 from the Government of India.

Conflict of Interest: none declared.

References

Alber,F. *et al.* (2007) The molecular architecture of the nuclear pore complex. **450**, 7.

Bhowmik,D. *et al.* (2018) Deep clustering of protein folding simulations. *BMC Bioinformatics*, **19**, 484.

Brilot,A.F. *et al.* (2021) CM1-driven assembly and activation of Yeast γ-Tubulin Small Complex underlies microtubule nucleation. *bioRxiv*, 2020.11.21.392803.

Cazals,F. and Tetley,R. (2019) Characterizing molecular flexibility by combining least root mean square deviation measures. *Proteins Struct. Funct. Bioinforma.*, **87**, 380–389.

Chen,S.H. *et al.* (2020) Distinct Structural Flexibility within SARS-CoV-2 Spike Protein Reveals Potential Therapeutic Targets. *bioRxiv*, 2020.04.17.047548.

Doerr,S. *et al.* (2017) Dimensionality reduction methods for molecular simulations. *ArXiv171010629 Cs Q-Bio Stat.*

Eguchi,R.R. *et al.* (2020) IG-VAE: Generative Modeling of Immunoglobulin Proteins by Direct 3D Coordinate Generation. *bioRxiv*, 2020.08.07.242347.
Ganesan,S.J. *et al.* (2020) Integrative structure and function of the yeast exocyst complex. *Protein Sci. Publ. Protein Soc.*, **29**, 1486–1501.
Gutierrez,C. *et al.* (2020) Structural dynamics of the human COP9 signalosome revealed by cross-linking mass spectrometry and integrative modeling. *Proc. Natl. Acad. Sci. U. S. A.*, **117**, 4088–4098.
Heinisch,O. (1962) Steel, R. G. D., and J. H. Torrie: Principles and Procedures of Statistics. (With special Reference to the Biological Sciences.) McGraw-Hill Book Company, New York, Toronto, London 1960, 481 S., 15 Abb.; 81 s 6 d. *Biom. Z.*, **4**, 207–208.
Hernández,C.X. *et al.* (2018) Variational encoding of complex dynamics. *Phys. Rev. E*, **97**, 062412.
Hinton,G.E. and Salakhutdinov,R.R. (2006) Reducing the Dimensionality of Data with Neural Networks. *Science*, **313**, 504–507.
Hoseini,P. *et al.* (2021) Generative deep learning for macromolecular structure and dynamics. *Curr. Opin. Struct. Biol.*, **67**, 170–177.
Jacobs,D.J. *et al.* (2001) Protein flexibility predictions using graph theory. *Proteins Struct. Funct. Bioinforma.*, **44**, 150–165.
Kedem,K. *et al.* (1999) Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories. *Proteins Struct. Funct. Bioinforma.*, **37**, 554–564.
Kim,S.J. *et al.* (2018) Integrative Structure and Functional Anatomy of a Nuclear Pore Complex. *Nature*, **555**, 475–482.
Kingma,D.P. and Ba,J. (2017) Adam: A Method for Stochastic Optimization. *ArXiv14126980 Cs*.
Kingma,D.P. and Welling,M. (2014) Auto-Encoding Variational Bayes. *ArXiv13126114 Cs Stat*.
Lasker,K. *et al.* (2012) Molecular architecture of the 26S proteasome holocomplex determined by an integrative approach. *Proc Natl Acad Sci USA*, **109**, 1380–1387.
Lee,H. *et al.* (2019) DeepDriveMD: Deep-Learning Driven Adaptive Molecular Simulations for Protein Folding. In, *2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*., pp. 12–19.
Liu,W. *et al.* (2020) Towards Visually Explaining Variational Autoencoders., pp. 8642–8651.
Luo,J. *et al.* (2015) Architecture of the human and yeast general transcription and DNA repair factor TFIIH. *Mol Cell*, **59**, 794–806.
Martínez,L. (2015) Automatic identification of mobile and rigid substructures in molecular dynamics simulations and fractional structural fluctuation analysis. *PloS One*, **10**, e0119264.
Matthews,B.W. (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta BBA - Protein Struct.*, **405**, 442–451.
Pfleger,C. *et al.* (2013) Global and local indices for characterizing biomolecular flexibility and rigidity. *J. Comput. Chem.*, **34**, 220–233.
Robinson,P. *et al.* (2015) Molecular architecture of the yeast Mediator complex. *eLife*, 10.7554/eLife.08719.
Romero,R. *et al.* (2019) Mechanism of glucocerebrosidase activation and dysfunction in Gaucher disease unraveled by molecular dynamics and deep learning. *Proc. Natl. Acad. Sci.*, **116**, 5086–5095.
Rout,M.P. and Sali,A. (2019) Principles for Integrative Structural Biology Studies. *Cell*, **177**, 1384–1403.

Russel,D. *et al.* (2012) Putting the pieces together: integrative structure determination of macromolecular assemblies. *PLoS Biol*, **10**, e1001244.
Sali,A. *et al.* (2015) Outcome of the First wwPDB Hybrid/Integrative Methods Task Force Workshop. *Structure*, **23**, 1156–67.
Saltzberg,D. *et al.* (2019) Modeling biological complexes using Integrative Modeling Platform. *Methods Mol Biol*, **2022**, 353–77.
Saltzberg,D.J. *et al.* (2021) Using Integrative Modeling Platform to compute, validate, and archive a model of a protein complex structure. *Protein Sci. Publ. Protein Soc.*, **30**, 250–261.
Schneidman-Duhovny,D. *et al.* (2014) Uncertainty in Integrative Structural Modeling. *Curr Opin Struct Biol*, **28**, 96–104.
Tsuchiya,Y. *et al.* (2019) Autoencoder-Based Detection of Dynamic Allostery Triggered by Ligand Binding Based on Molecular Dynamics. *J. Chem. Inf. Model.*, **59**, 4043–4051.
Vallat,B. *et al.* (2018) Development of a prototype system for archiving integrative/hybrid structure models of biological macromolecules. *Structure*, **26**, 894-904.e2.
Viswanath,S., Chemmama,I., *et al.* (2017) Assessing Exhaustiveness of Stochastic Sampling for Integrative Modeling of Macromolecular Structures. *Biophys J*, **113**, 2344–2353.
Viswanath,S., Bonomi,M., *et al.* (2017) The molecular architecture of the yeast spindle pole body core determined by Bayesian integrative modeling. *Mol Biol Cell*, **28**, 3298–3314.
Viswanath,S. and Sali,A. (2019) Optimizing model representation for integrative structure determination of macromolecular assemblies. *Proc. Natl. Acad. Sci.*, **116**, 540–545.
Ward,A.B. *et al.* (2013) Integrative Structural Biology. *Science*, **339**, 913–915.
Webb,B. *et al.* (2018) Integrative structure modeling with the Integrative Modeling Platform. *Prot Sci*, **27**, 245–258.
Wriggers,W. and Schulten,K. (1997) Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. *Proteins Struct. Funct. Bioinforma.*, **29**, 1–14.
Zhao,S. *et al.* (2018) InfoVAE: Information Maximizing Variational Autoencoders. *ArXiv170602262 Cs Stat*.