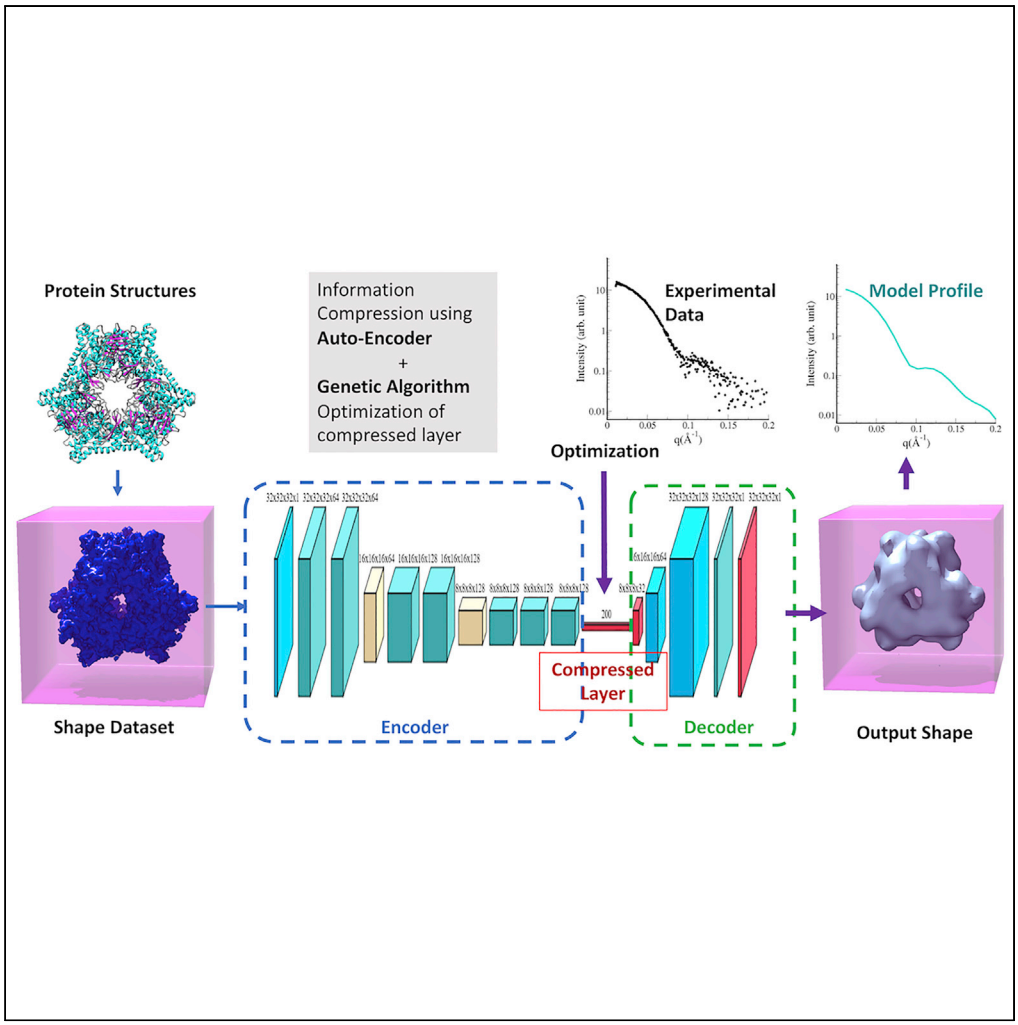
iScience

CellPress

Article

Model Reconstruction from Small-Angle X-Ray Scattering Data Using Deep Learning Methods



Hao He, Can Liu, Haiguang Liu

hgliu@csrc.ac.cn

HIGHLIGHTS

A convolutional neural network auto-encoder framework for 3D models is developed

The auto-encoder compresses protein shape information to 200 parameters

Accurate 3D models (both shape and radius) can be reconstructed from 1D SAXS data

He et al., iScience *23*, 100906 March 27, 2020
https://doi.org/10.1016/j.isci.2020.100906

iScience

CellPress

Article

Model Reconstruction from Small-Angle X-Ray Scattering Data Using Deep Learning Methods

Hao He,[1,2] Can Liu,[1,2] and Haiguang Liu[1,3,4,*]

SUMMARY

Small-angle X-ray scattering (SAXS) method is widely used in investigating protein structures in solution, but high-quality 3D model reconstructions are challenging. We present a new algorithm based on a deep learning method for model reconstruction from SAXS data. An auto-encoder for protein 3D models was trained to compress 3D shape information into vectors of a 200-dimensional latent space, and the vectors are optimized using genetic algorithms to build 3D models that are consistent with the scattering data. The program has been tested with experimental SAXS data, demonstrating the capacity and robustness of accurate model reconstruction. Furthermore, the model size information can be optimized using this algorithm, enhancing the automation in model reconstruction directly from SAXS data. The program was implemented using Python with the TensorFlow framework, with source code and webserver available from http://liulab.csrc.ac.cn/decodeSAXS.

INTRODUCTION

Small-angle X-ray scattering (SAXS) from protein molecules in solution is a powerful technique that provides information on molecular structures and dynamics (Grant et al., 2011; Putnam et al., 2007; Svergun and Koch, 2003). Because the solution scattering method does not require special treatment for protein molecules, such as crystallization in diffraction measurement or isotope labeling in nuclear magnetic resonance, SAXS experiments can be performed in high-throughput manners (Hura et al., 2009). Another major advantage of SAXS experiments is the ability to probe the structure and dynamics in solution, which is especially useful when combined with pumping methods to promote conformational changes (Kim et al., 2012; Neutze and Moffat, 2012). Time-resolved studies will reveal important information on molecular mechanism for protein functions.

Despite the success in extracting structural information from SAXS profiles, reconstructing high-quality 3D models remains challenging. Several approaches have been proposed and implemented to build 3D density maps from SAXS data, including shape envelope approximation using spherical harmonics functions, polymer chain folding, dummy atom assembly, iterative phasing, and database searching methods. The spherical harmonics function approximation method is fast but limited by resolution (Stuhrmann, 1970; Svergun and Stuhrmann, 1991; Svergun et al., 1996). In the Gasbor program, polymers composed of connected beads were used to represent protein molecules, and the packing of these polymers was optimized to build 3D models (Svergun et al., 2001). Dummy atoms arranged in a 3D lattice were also used for model reconstruction, as implemented in DAMMIN/DAMMIF (Franke and Svergun, 2009; Svergun, 1999). An iterative phase retrieval method was expanded to analyze SAXS data and demonstrated its applications (Grant, 2018). Using machine learning methods, Franke et al. developed a method to classify the shapes and gain valuable model parameters (Franke et al., 2018). There are also successful attempts to integrate SAXS data to molecular prediction/modeling/simulation frameworks to obtain 3D structures (Hura et al., 2019; Karczyńska et al., 2018). A database of shapes abstracted from actual protein complexes and efficiently represented using 3D Zernike polynomials was used to quickly retrieve 3D models that match experimental SAXS profiles, as implemented in sastbx.shapeup (Liu et al., 2012b). A real space representation of a 3D model requires many parameters, such as the position of each bead, which can be described using its coordinates (then 3N parameters are required for a model with N beads). However, the number of parameters required is much greater than the number of free parameters encoded in 1D SAXS profiles (Moore, 1980). Therefore, prior knowledge must be applied to provide additional constraints for converged reconstructions. For example, the molecular size and the connectivity of the beads are very critical for DAMMIN/DAMMIF. In the case of sastbx.shapeup, the molecular size is de-coupled from the abstracted shapes (Liu

[1]Complex Systems Division, Beijing Computational Science Research Center, 8 E Xibeiwang Road, Haidian, Beijing 100193, People's Republic of China

[2]School of Software Engineering, University of Science and Technology China, Suzhou, Jiang Su 215123, People's Republic of China

[3]Physics Department, Beijing Normal University, Haidian, Beijing 100875, People's Republic of China

[4]Lead Contact

*Correspondence: hgliu@csrc.ac.cn

https://doi.org/10.1016/j.isci.2020.100906

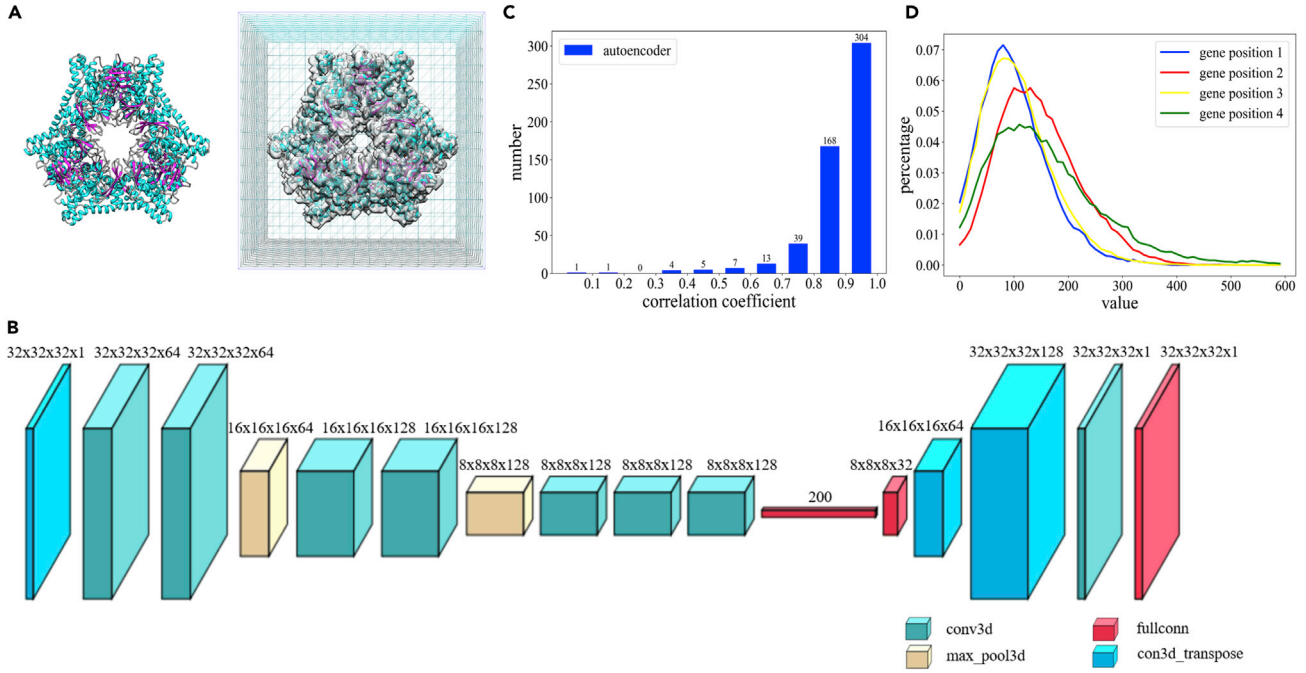


Figure 1. Framework of Auto-Encoder and Its Capability in Representing 3D Models

(A) The voxelization of a molecular structure. Left: an atomic model represented in the cartoon representation. Right: the model is mapped on a 3D matrix whose values are binarized depending on whether the grids are in the vicinity of any atom of the protein molecule.

(B) The auto-encoder-decoder architecture used in this study. The layers and structures are shown in the figure; details can be found in the Transparent Methods.

(C) The model quality encoded using the trained network and measured using the correlation coefficients between the models before and after going through the encode-decode process.

(D) The distribution of encoding parameter values for first four parameters in latent space.

et al., 2012a), allowing an optimization of the size as a separate parameter. However, the diversity of models is limited by the database. Model reconstruction will be significantly advanced if the following criteria are met: (1) diverse shapes of 3D models can be efficiently represented to cover a broader range than those in structure databases and (2) SAXS profiles can be computed for each model that can be scaled to arbitrary sizes. We provide a solution to achieve this using an auto-encoder method combined with 3D Zernike representations (Canterakis, 1999; Liu et al., 2012a).

Inspired by deep learning methods, real space 3D models were encoded using an auto-encoder neural network to a compressed representation of 200 latent parameters. The protein complexes in the PISA database (Krissinel and Henrick, 2007) were used to generate the training datasets for the auto-encoder. Each complex structure was scaled to the same radius (50Å was used in this study) then voxelized on a 3D grid of $31 \times 31 \times 31$ (i.e., voxel edge size is 50/15Å). Because SAXS data often provide low-resolution information that warrants a uniform density approximation for 3D models, we binarized the voxelized objects before auto-encoder training. Owing to the uniform density approximation of the reconstructed models, the SAXS data comparison were limited up to $q = 0.2$ Å^{-1} (Grant, 2018; Poitevin et al., 2011). The auto-encoder model is based on the VGG network (Simonyan and Zisserman, 2015), composed of convolutional, pooling, and full-connected layers (see Figure 1, Table S1 and the Transparent Methods). The testing results show that diverse shapes represented using 31^3 voxels with binary numbers can be accurately encoded using a vector of 200 dimensions. The reduction of parameter space allows application of optimization algorithms to improve the model-data fitting. SAXS profiles for 3D voxelized objects were computed using the Zernike expansion method, taking advantage of fast evaluation of theoretical profiles at an arbitrary model radius. The genetic algorithm (Goldberg, 1989) was used to optimize the latent space parameters, which were decoded to 3D models subsequently. One of the major advantages of the proposed method is the automatic determination of the model radius, which can be coded as an additional parameter and subject to the optimization along with the other 200 parameters. The testing results using experimental data from the

SASBDB (Valentini et al., 2015) and BIOISIS (Rambo and Tainer, 2011) show that the proposed method can successfully generate 3D models based on SAXS data. The algorithm is implemented to the software, *decodeSAXS*, whose source code and an associated webserver are available at http://liulab.csrc.ac.cn/decodeSAXS.

RESULTS

In this section, we first demonstrate that the auto-encoder neural network accurately represents the shape information in the compressed form. Then, we show the performance of model reconstruction with or without model size information using the SAXS data as the target for optimization. The performance was compared with other widely used model reconstruction algorithms. The model quality and consistency were also assessed by running multiple reconstructions and analyzing the similarity among reconstructed models. The reconstructions for experimental SAXS datasets yield high-quality 3D models in general with a few exceptions for challenging cases, such as loosely packed molecules or those with large cavities.

Quality and Accuracy of 3D Model Auto-Encoding

The voxelized objects derived from protein complex structures were encoded using 200 latent parameters in the auto-encoder neural network training procedure as described in the Transparent Methods. From two databases for small-angle scattering, the SASBDB and the BIOISIS, 542 SAXS datasets with deposited 3D models were obtained (see Supplemental Information for the full list). First, using the 3D models from these 542 datasets, the auto-encoder performance was evaluated. Each model was converted to a 3D voxel object with binary values and then fed to the trained auto-encoder for encoding and decoding. Then, each input model was used as the reference to assess the quality of the decoded 3D object. The real space correlation coefficient (denoted as *cc*, see Transparent Methods for detailed explanation) between original models and decoded models from the corresponding 200 latent parameters were computed and analyzed. The correlation coefficient is used to quantify the fraction of the overlapped volume compared with the geometrically averaged volume of two models. The auto-encoder network is very efficient and accurate in representing the majority of the 3D protein shapes, yielding a mean *cc* of 0.88, with 304 (56.1%) models having *cc* values greater than 0.90 (see Figure 1C). We also observed a few failed cases, whose correlation coefficients are very low. After investigating those failed encoding cases, we found that those models have very complex shapes, such as flexible chains (for instance SASDBD:SASDBZ6, with $cc = 0.40$) that have multiple conformations in the deposited dataset. For the majority of the testing models that have compact shapes, the auto-encoder works nicely in representing 3D shape information. The encoding-decoding test demonstrates that the 3D models are accurately reproduced after going through the encoding-decoding procedure, and the compressed 200-d vectors are sufficient to represent 3D molecular shapes. This lays the foundation for applying the auto-encoder method to reconstruct 3D models by optimizing compressed parameters to obtain models that fit to experimental SAXS data. Furthermore, the training dataset provides the distributions of latent variables, indicating that the valid values for these variables are distributed in limited ranges (see Figure 1D for distributions of representative variables). This prior information will facilitate the parameter sampling for the gene pool initialization and during the optimization process using genetic algorithms (see Figure S1 in Transparent Methods).

Performance of Reconstruction Algorithm with or without Model Size Information

The evaluation of the reconstruction algorithm was performed on the same 542 experimental datasets that were used to evaluate the performance of the auto-encoder neural network in the previous section. Here, we use the SAXS data as the only information to reconstruct the corresponding 3D models, and the deposited 3D structures along with the SAXS data are used as the references for model quality assessment after reconstruction (see Table S4).

The first reconstruction experiment was performed by assuming that the model size information is known. Model sizes can be derived from SAXS data using GNOM or other similar approaches (Liu and Zwart, 2012; Svergun, 1992). Instead of the maximum dimension obtained directly from the pairwise distance distribution functions, the auto-encoder method used the radius of minimal bounding sphere for the desired model (hereafter referred to as model radius). Here, the radius of deposited models (coarse-grained bead models or atomic models) from each SAXS dataset was used as the input radius for model reconstruction.

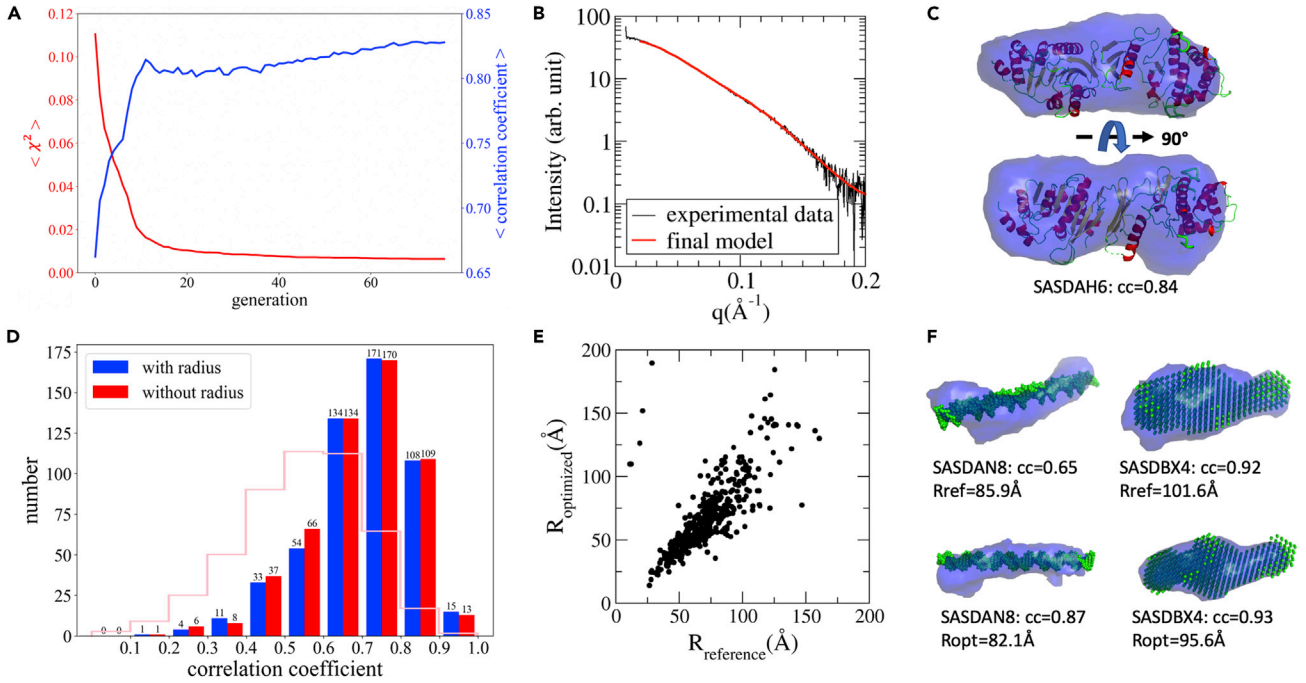


Figure 2. Performance of the *decodeSAXS* Algorithm

(A) The progress of model reconstruction by optimizing the goodness of fit to experimental data (SASBDB:SASDAH6). Chi-scores for SAXS data and the correlation coefficients for reconstructed models are shown for each iteration.

(B) The SAXS comparison with experimental data for the reconstructed model.

(C) The reconstructed model (blue surface) compared with the reference structure (obtained from SASBDB) at two orthogonal orientations.

(D) The algorithm performance measured using correlation coefficients between reconstructed models and the reference structures in the databases. Blue and red histograms show the statistics of correlation coefficients with or without using model size information as prior knowledge, respectively. The histogram represented with pink lines indicates the correlation coefficients for random paired models.

(E) The comparison between optimized model radii and the reference model radii.

(F) Representative reconstructions for two examples from the SASBDB with or without radius information.

The reconstruction process was monitored based on the chi-score between model SAXS profile and the experimental data. As a retrospective check, the reconstructed models are also compared with the reference model by computing their correlation coefficients after optimal alignment. Figures 2A–2C present an example to demonstrate the progress of model reconstruction. The dataset is from the SASBDB (SASDBD: SASDAH6), and the atomic structure deposited along with the SAXS data was used for model quality assessment. The chi-scores of SAXS data comparison and the *cc* from the 3D model comparison are shown for each iteration. As shown in Figure 2A, the chi-score was rapidly reduced within the first 10 iterations and gradually converged to a small value, indicating that the model SAXS profiles match the target SAXS data. Meanwhile, the correlation coefficients were improved as the model was iteratively reconstructed. The final model SAXS profile is shown in Figure 2B compared with the experimental data. The reconstructed model was superimposed onto the atomic model and shown in Figure 2C in two orthogonal orientations. The agreement between the reconstructed model (blue surface) and the atomic structure (cartoon model) illustrates the accuracy of the 3D model reconstruction for this dataset. The reconstruction process can also be directly visualized in real space by showing a sequence of reconstructed models in the form of videos (see Video S1 for an example). The progress clearly demonstrates that the genetic algorithm is capable of driving the model reconstruction by improving the goodness of fitting to experimental SAXS data.

Figure 2D shows the statistics of reconstructed model quality measured using correlation coefficients between the reconstructed models and the reference models. The histogram colored in blue shows reconstruction performance using the model radius as known information. As shown in the following, the radius information is not required for the reconstruction algorithm to achieve similar accuracy levels. Among 542 testing datasets, 294 reconstructed models have correlation coefficients greater than 0.70 (see Table S2 for

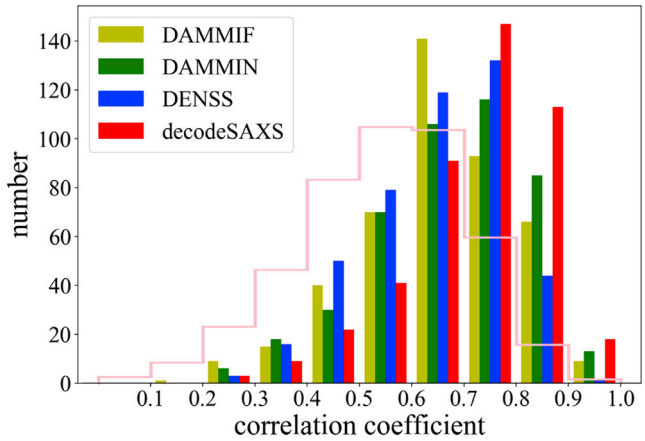


Figure 3. Performance Comparison with Other Methods

The histograms of the correlation coefficients between reconstructed models and reference models indicate that *decodeSAXS* outperforms the other three methods. The histogram with pink color shows the statistics of correlation coefficients between randomly selected models, providing a baseline to assess the performance of reconstruction algorithms.

representative models at various *cc* levels for visual inspections). At this level (*cc* = 0.70), the reconstructed models are consistent with the references in the overall shapes according to visual inspections. The models with a big cavity or flexible domains are challenging for the auto-encoder to compress the shape information to a 200-dimension vector, as discussed in the previous section. For models with rigid and compact structures, the auto-encoder and the SAXS-based reconstruction are very successful. Furthermore, random pairing the models was used as the control method, and the *cc* between randomly selected models were used to estimate a baseline for reconstruction performance assessment. Prior to the alignment and correlation coefficient computation, two models randomly selected from the 542 reference models were scaled to the same size. This procedure was repeated for 100,000 times to get the statistics of the *cc* values, which are scaled and shown with pink lines in Figure 2D. The average *cc* value of the control method is 0.55 with a standard deviation of 0.16, suggesting that the expectation value of *cc* is 0.55 by random matching. The *decodeSAXS* algorithm performed much better than random matching. Among 542 reconstructed models, 86% showed *cc* > 0.55 and 54% models had *cc* > 0.70 in the case of the *decodeSAXS* using correct model radius.

Three other methods, DAMMIN, DAMMIF, and DENSS, were applied to the same dataset for model reconstructions. To reduce the bias during model reconstructions, default parameters and correct radii were used for model building in all programs. The reconstructed model quality was measured using the correlation coefficients and compared with the *decodeSAXS* method (Figure 3). The results revealed that all reconstruction methods generated models better than randomly selected models, whose statistics was used as a reference (pink line in Figure 3). Detailed analysis showed that the performance of DAMMIN was better than DAMMIF in general, whereas DENSS achieved similar accuracy as DAMMIN. Based on the statistics, *decodeSAXS* program outperformed these three methods on the testing dataset, reflected by larger populations with higher *cc* values (red color histogram in Figure 3). The scattering plots of *cc* values between references models and those reconstructed using either *decodeSAXS* or the other methods are summarized in Figure S3, showing better performance of *decodeSAXS* in detailed comparisons.

In practice, the size information of the model to be reconstructed could not be accurately obtained in many cases, preventing successful model reconstructions. In such cases, the *decodeSAXS* algorithm can optimize the model size under the same framework by simply treating the model radius as an additional element in the genes (the parameters). Using the same dataset, the reconstruction algorithm was tested without providing the size information. The initial radius for each model in the first generation is a random positive number smaller than 300 (with the associated unit Å). The radius was taken as the 201st parameter and subject to the genetic algorithm for optimization. The optimized radii for 542 testing datasets are compared with the radii extracted from the reference models in Figure 2E, showing that the size

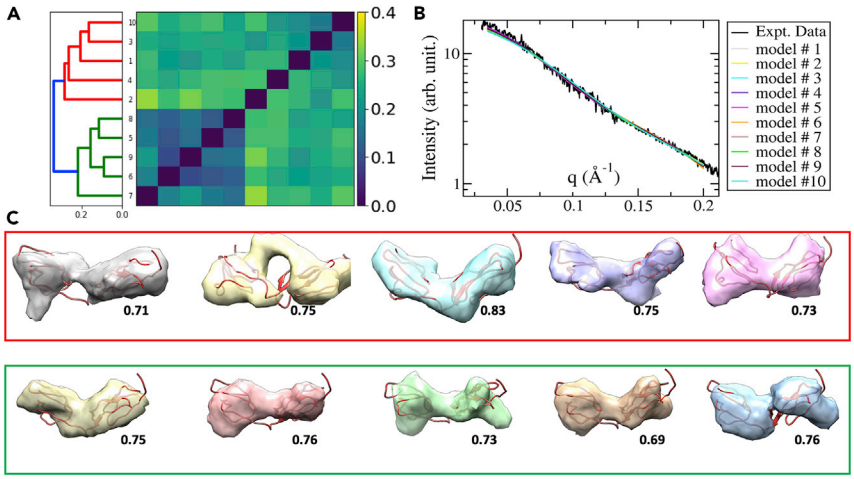


Figure 4. Reconstruction Model Quality and Consistency

(A) The clustering analysis of 10 reconstructed models for dataset SASDA25. Using the correlation distance cutoff of 0.3, 10 reconstructed models were classified to two groups (see [C] for the reconstructed models superposed to the atomic structure of the same protein).

(B) The theoretical SAXS profiles of reconstructed models were fitted to the experimental data.

(C) The 10 reconstructed models grouped into two classes, enclosed by the red and green boxes, corresponding to the two classes in (A). The correlation coefficients compared with the PDB structure are indicated next to each reconstructed model.

information can be obtained by optimization. More importantly, the reconstructed model quality without using the correct model radius is comparable with the outcomes with radius information as shown in Figure 2D (red color histogram for the cases with optimized radius, and the blue color histogram is for the case with correct radius, whereas the pink lines show the statistics for the comparison between randomly paired models). Two examples shown in Figure 2F demonstrate that high-quality 3D models can be reconstructed even if the radius is not exactly the same as the values of reference models, indicating the robustness of the algorithm. The automated determination of model radius during reconstruction process is an important feature of the *decodeSAXS* algorithm.

Consistency of Model Reconstructions

Because of the random initialization of the genetic algorithm, multiple reconstructions for the same SAXS dataset may lead to different models. Furthermore, the limited information contents in 1D SAXS profiles may also result in multiple 3D models that match to the experimental SAXS profile equally well. To test the model consistency, eight SAXS datasets were randomly chosen for multiple reconstructions. For each SAXS dataset, ten models were reconstructed by running the programs ten times with different initial models. Then the hierarchical clustering analyses were carried out using the correlation derived distance (see Transparent Methods). The clustering results indicate that single clusters were obtained in six of eight cases (see Table S3 for the clustering plots and the distance matrices). In the other two cases, the ten models were clustered into two classes (Figure 4A, as an example for dataset SASDBD:SASDA25). The computed model profiles in Figure 4B suggest similar levels of agreement to the experimental SAXS data. The reconstructed models for dataset SASDA25 are shown in Figure 4C, each superposed to the high-resolution PDB model. The models are grouped and enclosed using red and green boxes to indicate their classifications. The visual comparison and computed correlation coefficients compared with the PDB model both indicate that reconstructions are consistent with the reference PDB model (with $cc > 0.70$).

DISCUSSIONS AND CONCLUSION

3D model reconstruction from 1D SAXS data is challenging given the limited information embedded in the 1D profile. Prior knowledge, especially size information, has been required for model reconstruction. Here, using the deep learning method, the 3D models can be compressively represented using 200-dimension vectors. Based on the success in dimension reduction using the auto-encoder neural network methods, a model reconstruction method is implemented by optimizing parameters in the 200-d latent parameter

space. More importantly, this new method does not require model size information and demonstrates its robustness in 3D model reconstructions. Currently, the 200-d vector and the SAXS profile are not directly related but related via the decoding part of the neural network to get the corresponding 3D model and subsequent calculate its SAXS profile. This work demonstrates that the deep learning methods have potential applications in the interpretation of X-ray data, with properly designed interfaces between the model and X-ray data. The calculation speed may not have clear advantages over the methods that construct 3D models in real space, owing to the extra decoding procedure. Nevertheless, the proposed method opens up a new avenue for model reconstruction. It is possible to improve the efficiency and accuracy of model reconstructions, by expanding the approach presented in this work. An ultimate method is to compress the latent space to match the SAXS data, so that the trained neural network can directly "translate" the SAXS information to 3D models. In such frameworks, there are information inputs from both 3D models and 1D SAXS data, and to train such neural networks, more complicated neural network architecture is desired. In a separate study, preliminary results show the possibility to encode a SAXS profile using low-dimensional vectors rather than computing from the decoded 3D models, and this feature will significantly reduce the computing time. Linking the 200-D parameter in this study and the latent space vector for SAXS data encoding will be another alternative toward a *de novo* model reconstruction method without using iterative model building. On the other hand, the present approach might be slow, but it can be easily expanded to other applications with an interface to convert decoded 3D models to experimental measurable information (SAXS profile in this case).

Neural network training and application can benefit from parallel processing of GPU; the speed gained from advanced hardware can facilitate the multiple reconstructions for model consistency examination. The computing time for model reconstruction is not the bottleneck in SAXS studies, considering that the sample preparation and experimental execution usually take much longer time. Furthermore, the auto-encoder is not limited to represent uniform density models; therefore, it is possible to expand the neural network to encode models with density variations from the uniform density approximation models.

The progression of latent space parameters showed large variations of parameter values at the beginning stage of the optimization, and these parameters were converged to fixed values when the optimization progressed to 60–80 iterations. The linear interpolation between initial random values and the final optimized values can be decoded to 3D models as "intermediate models." The interpolation of latent space parameters resulted a morphing of the decoded 3D models, from a roundish object to a model that matches SAXS data, revealing the relation between latent space parameters and real space models (see Video S2 for a demonstration of model morphing using latent parameter interpolation).

As the first demonstration for the deep learning method applied in the 3D model reconstruction from SAXS data, this work may open doors for potential applications of deep learning in understanding other types of experimental data, especially those that are insufficient to be directly converted to 3D models. For instance, imaging experimental data with one or a few 2D projection or scattering information could be analyzed in a similar approach to build 3D models. This method has demonstrated its capacity and robustness in building 3D models with diverse shapes from experimental data. As more high-throughput X-ray data are being collected, we anticipate increasing applications of such methods in data analysis.

Limitations of Study

Owing to the low information content embedded in SAXS data, the present method is limited to the reconstruction of models with uniform density. In principle, the method can be expanded to reconstruct higher-resolution models to reflect density variations within molecular envelopes. This shall require more advanced neural network architecture to encode models with density variations.

The maximum scattering angle used for model reconstruction is 0.2 $Å^{-1}$ in this study. Proper modeling of the hydration layer at molecular surfaces is desired to utilize the scattering signals at larger scattering angles.

METHODS

All methods can be found in the accompanying Transparent Methods supplemental file.

DATA AND CODE AVAILABILITY

The data and program codes are available at http://liulab.csrc.ac.cn/decodeSAXS or https://github.com/LiuLab-CSRC/SAXS_reconstruction.

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.isci.2020.100906.

ACKNOWLEDGMENTS

Funding from National Natural Science Foundation of China (grant numbers: 11575021, U1530402, U1430237) is acknowledged.

AUTHOR CONTRIBUTIONS

Conceptualization, H.L.; Methodology, H.H., C.L., and H.L.; Software, H.H., C.L., and H.L.; Visualization, H.H. and H.L.; Writing – Original Draft, H.H., C.L., and H.L.; Writing – Review & Editing, H.H and H.L.; Funding Acquisition, H.L.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: May 8, 2019
Revised: November 18, 2019
Accepted: February 7, 2020
Published: March 27, 2020

REFERENCES

Canterakis, N. (1999). 3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition. 11th Scand. Conf Image Anal.

Franke, D., and Svergun, D.I. (2009). DAMMIF, a program for rapid ab-initio shape determination in small-angle scattering. J. Appl. Crystallogr. *42*, 342–346.

Franke, D., Jeffries, C.M., and Svergun, D.I. (2018). Machine learning methods for X-ray scattering data analysis from biomacromolecular solutions. Biophys. J. *114*, 2485–2492.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning (Reading, MA: Addison-Wesley).

Grant, T.D. (2018). Ab initio electron density determination directly from solution scattering data. Nat. Methods *15*, 191–193.

Grant, T.D., Luft, J.R., Wolfley, J.R., Tsuruta, H., Martel, A., Montelione, G.T., and Snell, E.H. (2011). Small angle X-ray scattering as a complementary tool for high-throughput structural studies. Biopolymers *95*, 517–530.

Hura, G.L., Menon, A.L., Hammel, M., Rambo, R.P., Poole, F.L., Tsutakawa, S.E., Jenney, F.E., Classen, S., Frankel, K.A., Hopkins, R.C., et al. (2009). Robust, high-throughput solution structural analyses by small angle X-ray scattering (SAXS). Nat. Methods *6*, 606–612.

Hura, G.L., Hodge, C.D., Rosenberg, D., Guzenko, D., Duarte, J.M., Monastyrskyy, B., Grudinin, S., Kryshtafovych, A., Tainer, J.A., Fidelis, K., et al. (2019). Small angle X-ray scattering-assisted protein structure prediction in CASP13 and emergence of solution structure differences. Proteins Struct. Funct. Bioinf. *87*, 1298–1314.

Karczyńska, A.S., Mozolewska, M.A., Krupa, P., Giełdoń, A., Liwo, A., and Czaplewski, C. (2018). Prediction of protein structure with the coarse-grained UNRES force field assisted by small X-ray scattering data and knowledge-based information. Proteins Struct. Funct. Bioinf. *86*, 228–239.

Kim, K.H., Muniyappan, S., Oang, K.Y., Kim, J.G., Nozawa, S., Sato, T., Koshihara, S., Henning, R., Kosheleva, I., Ki, H., et al. (2012). Direct observation of cooperative protein structural dynamics of homodimeric hemoglobin from 100 ps to 10 ms with pump-probe X-ray solution scattering. J. Am. Chem. Soc. *134*, 7001–7008.

Krissinel, E., and Henrick, K. (2007). Inference of macromolecular assemblies from crystalline state. J. Mol. Biol. *372*, 774–797.

Liu, H., and Zwart, P.H. (2012). Determining pair distance distribution function from SAXS data using parametric functionals. J. Struct. Biol. *180*, 226–234.

Liu, H., Morris, R.J., Hexemer, A., Grandison, S., and Zwart, P.H. (2012a). Computation of small-angle scattering profiles with three-dimensional Zernike polynomials. Acta Crystallogr. A *68*, 278–285.

Liu, H., Hexemer, A., and Zwart, P.H. (2012b). The Small Angle Scattering ToolBox (SASTBX): an open source software for biomolecular small angle scattering. J. Appl. Crystallogr. *45*, 587–593.

Moore, P.B. (1980). Small-angle scattering. Information content and error analysis. J. Appl. Crystallogr. *13*, 168–175.

Neutze, R., and Moffat, K. (2012). Time-resolved structural studies at synchrotrons and X-ray free electron lasers: opportunities and challenges. Curr. Opin. Struct. Biol. *22*, 651–659.

Poitevin, F., Orland, H., Doniach, S., Koehl, P., and Delarue, M. (2011). AquaSAXS: a web server for computation and fitting of SAXS profiles with non-uniformally hydrated atomic models. Nucleic Acids Res. *39*, W184–W189.

Putnam, C.D., Hammel, M., Hura, G.L., and Tainer, J.A. (2007). X-ray solution scattering ({SAXS}) combined with crystallography and computation: defining accurate macromolecular structures, conformations and assemblies in solution. Q. Rev. Biophys. *40*, 191–285.

Rambo, R.P., and Tainer, J.A. (2011). Characterizing flexible and intrinsically unstructured biological macromolecules by SAS using the Porod-Debye law. Biopolymers *95*, 559–571.

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In 3rd International Conference on Learning Representations, ICLR 2015-Conference Track Proceedings.

Stuhrmann, H.B. (1970). Improvements in the shape determination method based on multipole expansion. Z. Phys. Chem. Frankfurt *72*, 177–184, 185–198.

Svergun, D.I. (1992). Determination of the regularization parameter in indirect-transform

methods using perceptual criteria. J. Appl. Crystallogr. *25*, 495–503.

Svergun, D.I. (1999). Restoring low resolution structure of biological macromolecules from solution scattering using simulated annealing. Biophys. J. *76*, 2879–2886.

Svergun, D.I., and Koch, M.H.J. (2003). Small-angle scattering studies of biological macromolecules in solution. Rep. Prog. Phys. *66*, 1735–1782.

Svergun, D.I., and Stuhrmann, H.B. (1991). New developments in direct shape determination from small-angle scattering. 1. Theory and model calculations. Acta Crystallogr. Sect. A *47*, 736–744.

Svergun, D.I., Volkov, V.V., Kozin, M.B., and Stuhrmann, H.B. (1996). New developments in direct shape determination from small-angle scattering. 2. Uniqueness. Acta Crystallogr. Sect. A Found. Crystallogr. *52*, 419–426.

Svergun, D.I., Petoukhov, M.V., and Koch, M.H. (2001). Determination of domain structure of proteins from X-ray solution scattering. Biophys. J. *80*, 2946–2953.

Valentini, E., Kikhney, A.G., Previtali, G., Jeffries, C.M., and Svergun, D.I. (2015). SASBDB, a repository for biological small-angle scattering data. Nucleic Acids Res. *43*, D357–D363.

iScience, Volume *23*

Supplemental Information

Model Reconstruction from Small-Angle X-Ray Scattering Data Using Deep Learning Methods

Hao He, Can Liu, and Haiguang Liu

Table S1. The detailed configuration for each layer, related to Figure 1b.

layer	parameters	shape
Input:		In:(32,32,32,1)
conv1_1	k= (3,3,3,1,64) strides = (1,1,1,1,1) padding =“SAME ”	In:(32,32,32,1) out:(32,32,32,64)
conv1_2	filter = (3,3,3,64,64) strides = (1,1,1,1,1) padding = “SAME ”	In:(32,32,32,64) Out:(32,32,32,64)
relu		
max_pool1	ksize = (1,2,2,2,1) strides = (1,2,2,2,1)	In:(32,32,32,64) out:(16,16,16,64)
conv2_1	filter = (3,3,3,64,128) strides = (1,1,1,1,1) padding =“SAME ”	In:(16,16,16,64) out:(16,16,16,128)
conv2_2	filter = (3,3,3,128,128) strides = (1,1,1,1,1) padding =“SAME ”	In:(16,16,16,128) out:(16,16,16,128)
relu		
max_pool2	ksize = (1,2,2,2,1) strides = (1,2,2,2,1)	In:(16,16,16,128) out:(8,8,8,128)
conv3_1	filter = (3,3,3,128,128) strides = (1,1,1,1,1) padding =“SAME ”	In:(8,8,8,128) out:(8,8,8,128)
conv3_2	filter = (3,3,3,128,128) strides = (1,1,1,1,1) padding =“SAME ”	In:(8,8,8,128) out:(8,8,8,128)
conv3_3	filter = (3,3,3,128,128) strides = (1,1,1,1,1) padding =“SAME ”	In:(8,8,8,128) out:(8,8,8,128)
relu		
fc1		In:(8,8,8,128) Out:(200)
relu		
fc2		In:(200) Out:(8,8,8,128)
relu		
deconv1	filter = (5,5,5,64,32) strides = (1,2,2,2,1) padding =“SAME ”	in:(8,8,8,32) out:(16,16,16,64)
relu		
deconv2	filter = (5,5,5,128,64) strides = (1,2,2,2,1) padding =“SAME ”	in:(16,16,16,64) out:(32,32,32,128)
relu		
conv4	filter = (3,3,3,128,1) strides = (1,1,1,1,1) padding =“SAME ”	In:(32,32,32,128) out:(32,32,32,1)
sigmoid		

conv: convolution layer; **fc**: fully-connected laser; **max_pool**: pooling layer; **relu**: activation operator; **deconv**: deconvolution layer.
Gray shaded layers are for the encoder part, and the blue shaded layers are for the decoder part, respectively.

The Tensorflow framework is used for the neural network implementation and training.

The encoder part is composed of seven convolutional layers and two pooling layers, connected with Relu activation functions, and the last layer of the encoder part is a fully connected layer.

The decoder part includes a fully connected layer, followed by two cov3d_transpose layers, and concluded with a conv3d layer. It also utilizes Relu activation functions between layers and a sigmoid function to obtain the final output.

In the auto-encoder network, Relu activation function is applied after conv1_2, conv2_2, conv3_3, fc1, fc2, deconv1, deconv2. The pooling method used in the encode part is max-pooling with padding, and the stride is 2 in 3d space.

The decoder part utilizes conv3d_transpose with padding to increase its size, and the stride is 2 in 3d space.

Key parameters used for training:

BATCH_SIZE = 64
LEARNING_RATE = 0.01 , 0.001 or exponential_decay
TRAINING_EPOCHS = 15

The optimizer function is AdamOptimizer, and the cross entropy was used as the loss function.

Two-stage network training

The auto-encoder network model was trained in two stages, with shapes in the PISA database as the training dataset, which contains 60,000 samples.

Stage-1:

Setting the linking layer between encoder and decoder (i.e., the fc1-output and fc2-input, or the bottleneck layer) to be 3,000 (i.e., the 3D models are encoded to 3,000-D vectors), and other parameters are listed in Table S1. When the training result converges, the preliminary auto-encoder model is saved for the second stage training.

Stage-2:

Based on the auto-encoder model trained in Stage-1, the linking layer size is reduced to 200, and re-train the fc1 and fc2 layers **only**, while retaining other parameters in the rest of the auto-encoder network. The final trained network will have 200 parameters in this linking layer for 3D shapes.

Alternatively, the training can also be accomplished with a single stage training network by inserting the 200d vector layer to the first stage model, right after the layer of 3000 variables. These two layers are linked using a fully connected network. The two approaches yielded similar results. The source codes for both approaches are available from the Github repository.

Table S2. Representative reconstructed models compared to the reference models, related to Figure 2.

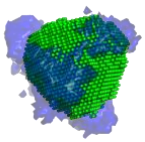
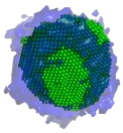
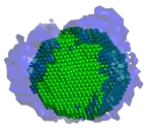
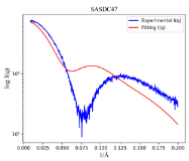
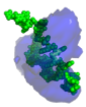
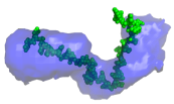
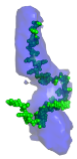
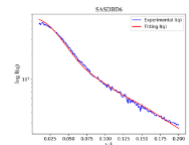
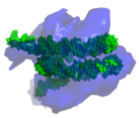
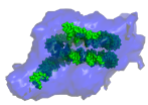
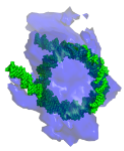
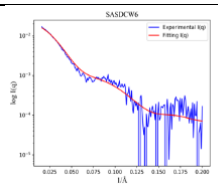
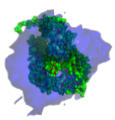
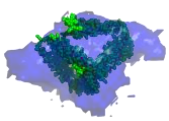
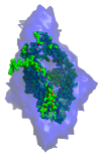
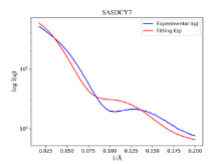
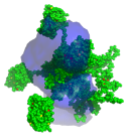
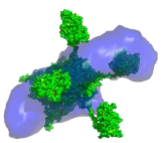
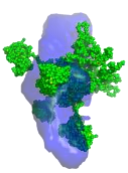
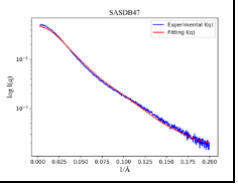
*Rref: Radius of reference model deposited to the database (in the unit of Å)

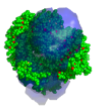
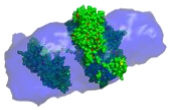
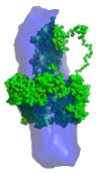
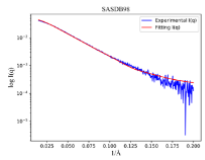
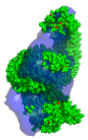
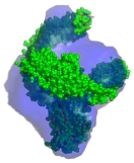
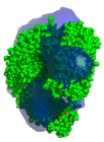
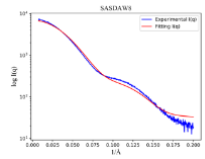
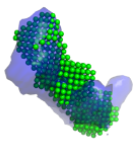
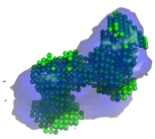
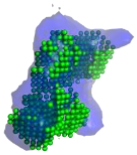
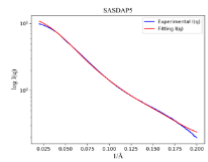
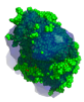
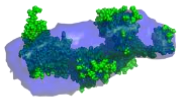
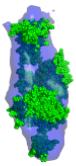
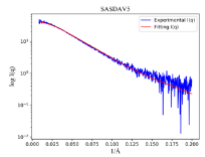
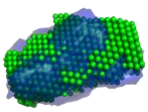
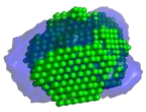
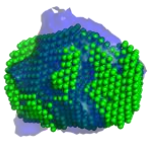
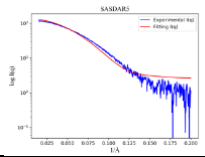
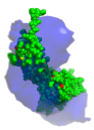
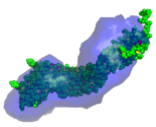
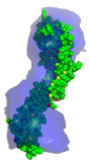
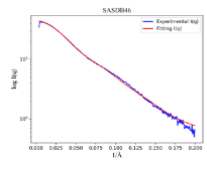
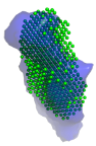
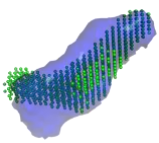
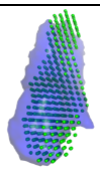
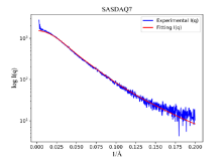
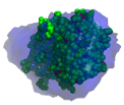
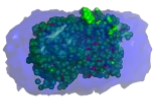
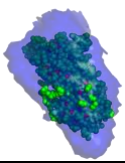
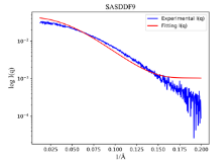
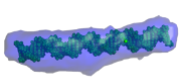
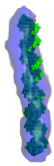
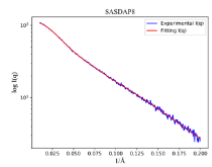
#Ropt: Radius optimized using the genetic algorithm in the *decodeSAXS* program (in the unit of Å)

To illustrate the model reconstruction accuracy at various correlation coefficient (*cc*) levels (in ascending order), reconstructed models are superimposed to the models deposited to the database. The deposited models are either coarse grain bead models or high resolution atomic models.

Based on visual comparison, the model quality is very good if the cc is greater than 0.70. For the models with low cc values, major causes are that the protein complexes are very dynamic and flexible, or loosely packed (see SASDB47, SASDB98). For SASDCW6, the SAXS data were collected with a buffer matching method to mask the histone protein component, so that the DNA superhelices were the only 'visible' component for X-rays. This is very unusual for biomolecules, so the *decodeSAXS* failed in 3D model reconstructions for this dataset. For SASDC47, the optimized radius (Rmax) was wrong, resulting a wrong model.

As shown in the main text, using cc=0.70 as a threshold, about 54% reconstructions are in good quality; if the cc threshold is relaxed to 0.55 (average *cc* for randomly paired models) in this dataset, then about 86% reconstructions can be considered to be successful. The performance does not rely on the prior knowledge of the molecular sizes, and this is a unique feature of this method.

Data ID	*cc*	Rref*/ Ropt#	View-1	View-2	View-3	Fitting I(q)
SASDC 47	0.14	59.19/ 85.90				
SASDB D6	0.41	53.72/ 64.60				
SASDC W6	0.43	88.72/ 111.85				
SASDC Y7	0.45	57.17/ 84.45				
SASDB 47	0.47	84.38/ 103.05				

Data ID	*cc*	Rref*/ Ropt#	View-1	View-2	View-3	Fitting I(q)
SASDB98	0.48	70.60/ 89.30				
SASDAW8	0.57	70.78/ 78.25				
SASDAP5	0.67	48.58/ 58.60				
SASDAV5	0.68	75.21/ 78.50				
SASDAR5	0.71	40.35/ 52.05				
SASDB46	0.73	74.51/ 70.25				
SASDAQ7	0.79	87.67/ 90.65				
SASDDF9	0.80	35.94/ 51.95				
SASDAP8	0.84	69.74/ 77.85				

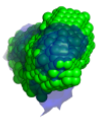
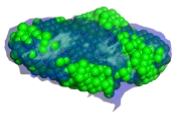
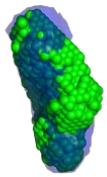
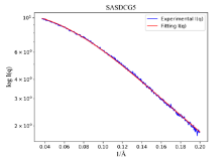
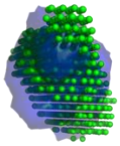
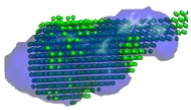
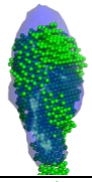
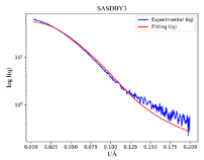
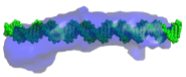
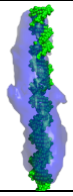
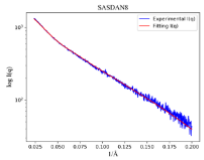
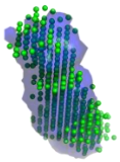
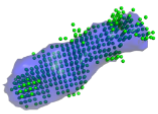
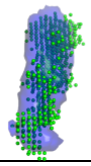
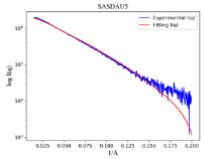
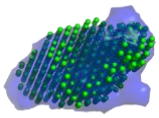
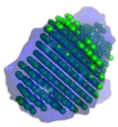
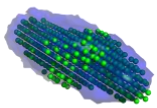
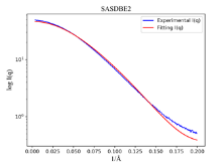
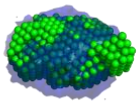
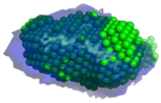
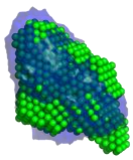
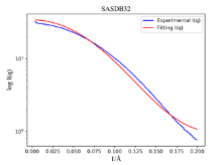
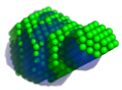
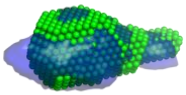
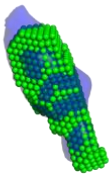
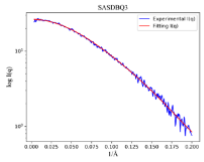
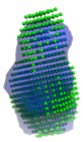
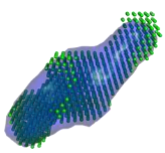
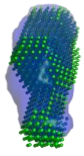
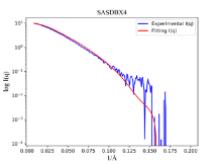
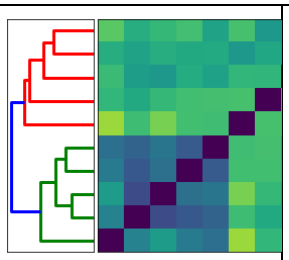
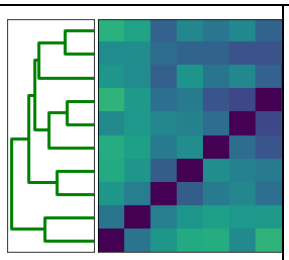
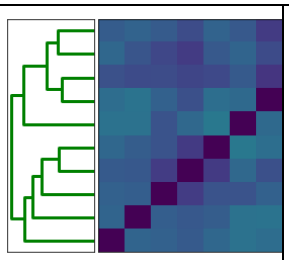
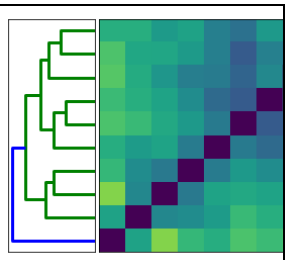
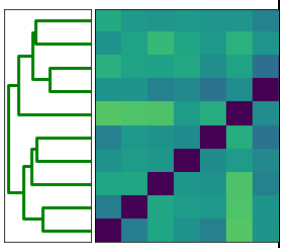
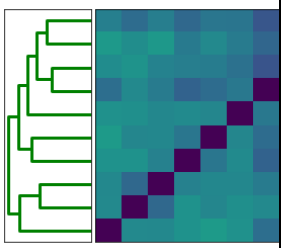
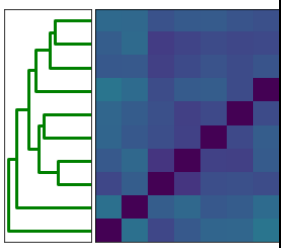
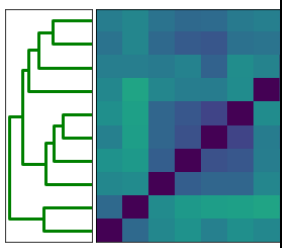
Data ID	*cc*	Rref*/ Ropt#	View-1	View-2	View-3	Fitting I(q)
SASDCG5	0.85	24.63/ 27.75				
SASDBY3	0.85	82.28/ 69.10				
SASDAN8	0.87	85.90/ 82.05				
SASDAU5	0.87	83.48/ 82.30				
SASDBE2	0.87	46.95/ 53.55				
SASDB32	0.88	37.37/ 39.00				
SASDBQ3	0.91	48.44/ 46.15				
SASDBX4	0.93	101.57 /95.60				

Table S3. Hierarchical clustering analysis of multiple reconstructions for 8 tested datasets, related to Figure 4.

Eight SAXS datasets were randomly selected for multiple reconstruction tests. The correlation between reconstructed models were computed. The hierarchical clustering method was applied using the distance metric defined as d=1.0 – cc, where cc is the correlation coefficient between reconstructed models calculated using sastb.superpose. Using d=0.3 as cutoff (i.e., correlation coefficient=0.7), 6 out of 8 testing cases showed single cluster; and 2 cases (SASDA25 & SASDA28) showed two clusters.

			
SASDA25	**SASDA26**	**SASDA27**	**SASDA28**
			
SASDA32	**SASDA35**	**SASDA38**	**SASDA39**

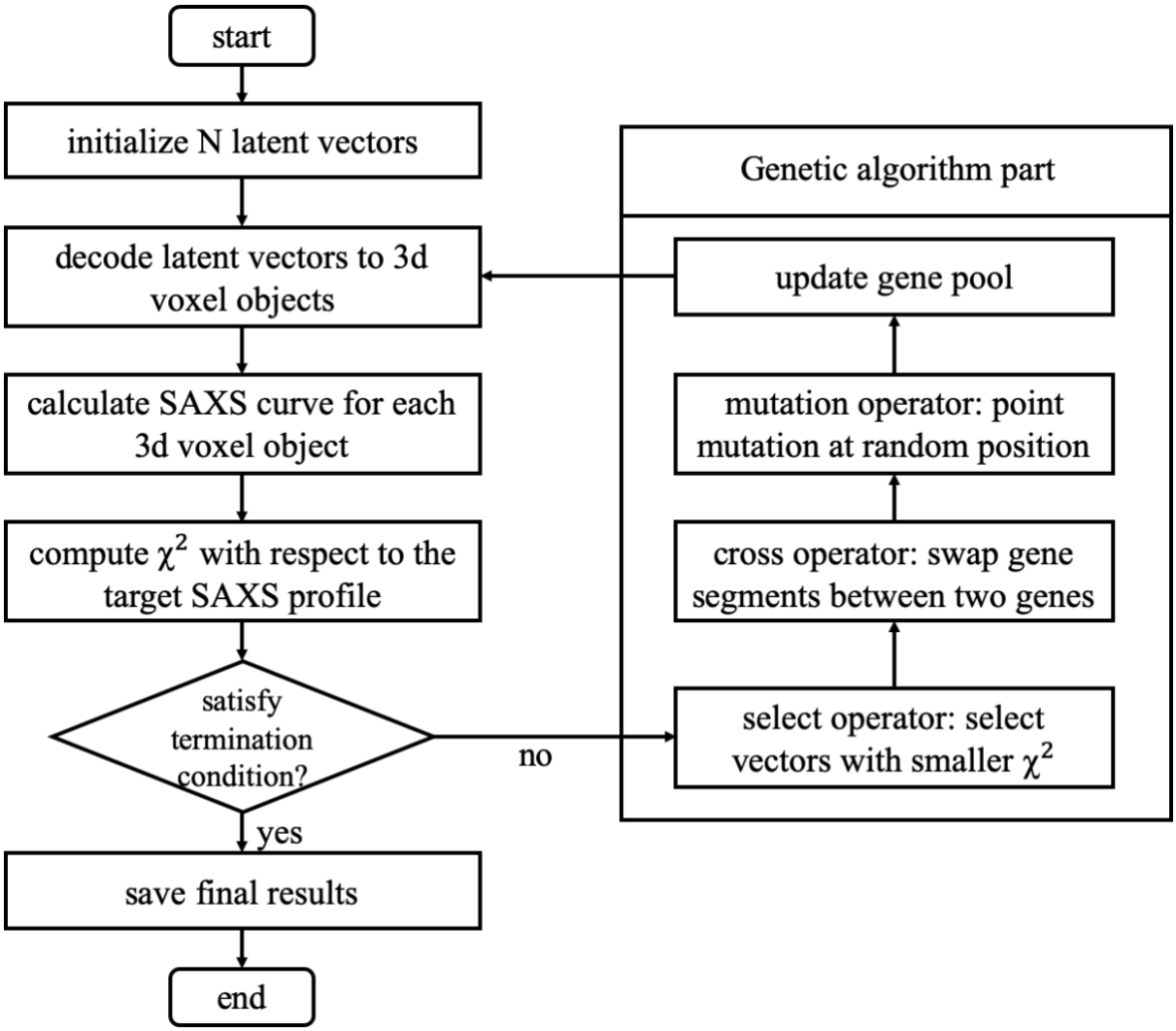


Figure S1. The flowchart of model reconstruction algorithm, related to Figure 2. The auto-encoder-decoder neural network is used to decode latent space parameter to 3D voxel models, whose SAXS profiles are compared to experimental data. Genetic algorithm is used to guide the optimization of latent space parameters.

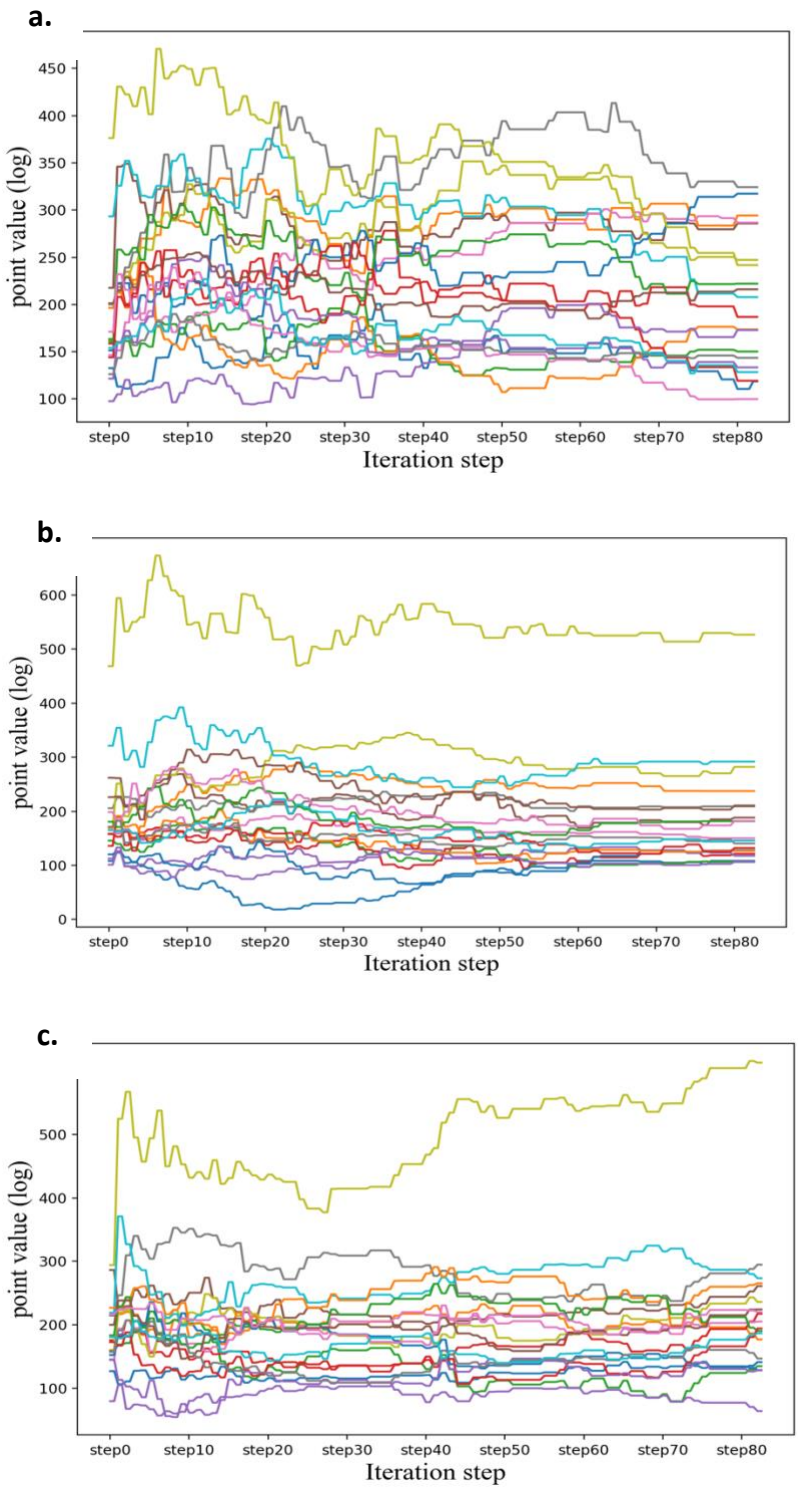


Figure S2. The progression of the first 20 parameters in the latent space, related to Figure 2. The parameters widely vary at the beginning and the convergence start to emerge after some iterations. Three examples are shown as (**a**) SASDA38; (**b**) SASDBQ3; and (**c**) SASDBY3.

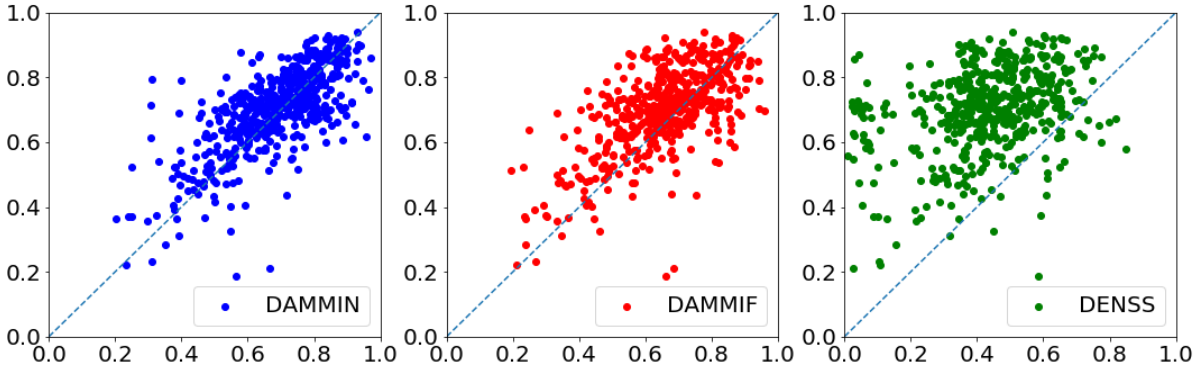


Figure S3. The performance comparison between decodeSAXS and three other methods, related to Figure 3. The scatter plots are shown for the cc values: y-axis shows the cc values between reference models and the models reconstructed using decodeSAXS; x-axis shows the cc values between reference models and the models reconstructed using the three other methods (labelled in the legends). The points above the line (y=x) corresponding to the cases that decodeSAXS performs better.

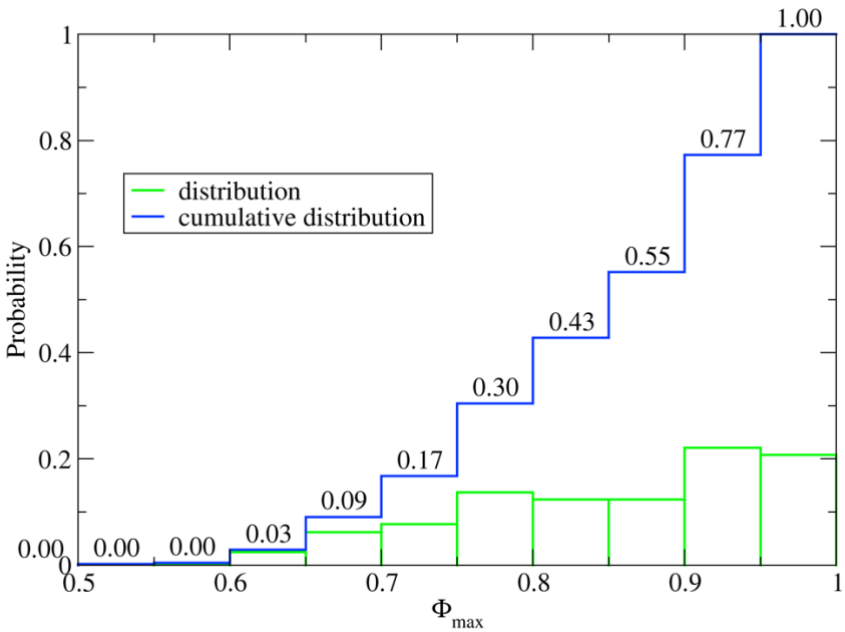


Figure S4. The distribution of Phi$_{max}$ values, related to Figure 2. The values are computed between the reconstructed models with decodeSAXS and the reference models. The distribution (green histogram) shows that Phi$_{max}$ has a larger population at larger values, compared to the cc values (Figure 2d). The cumulative distribution (blue line) shows that 70% of the Phi$_{max}$ values are larger than 0.80.

Transparent Methods

Training and Testing Datasets

The model dataset is compiled from the PISA structure database, including 60,000 randomly selected 3D models. Each model was first scaled and shifted to fit in a sphere centred at the coordinate origin with a radius of 50 Å (the shape does not depend on the size of the model under the uniform density model approximation; the size information is used for SAXS profile computation, see equation 2 below). Then, the atomic positions were mapped to a grid of 31x31x31 in the process of scaling and voxelization (the grid point at (0,0,0) coincides with the center of mass). As a result, each model was converted to a voxel object described using a 3D matrix with binary values (i.e., the uniform density is ensured, see Figure 1a in main text). For numerical calculation efficiency consideration, the matrix of 31x31x31 was padded with zeros to a matrix of 32x32x32, such that the number of discretization is a power of 2, to facilitate the convolutional neural network training.

Auto-encoder Neural network architecture

The architecture of the auto-encoder is designed based on the VGG network (Simonyan and Zisserman, 2015). The encoding part of the auto-encoder is composed of seven convolution layers and two pooling layers followed by one dense (fully connected) layer as indicated in Figure 1b and described in Table S1. Network training was performed in two stages. During the first stage, the dense layer contains 3,000 variables, and this number is reduced to 200 during the second stage. With this design, the 3D shape information is compressed to a representation of 3,000-dimensional vectors after the first training stage. Among these 3,000 parameters, a significant portion (approximately 90%) of parameters is found to be zero persistently, indicating that the parameter space for encoding can be further reduced. With this observation, the fully connected layer was optimized again with a reduced dense layer of 200 parameters. During the second stage of training, the parameters for convolutional and pooling layers were inherited from the first stage and remained unchanged, except that the parameters for the fully connected dense layer were subjected to optimization. This two-stage training was adapted to ensure fast convergence of the training (we found that if the dense layer was set to a 200-dimension vector during the first stage of training, the loss function does not converge). There is an alternative architecture to allow the training to be completed within a single stage, by adding the 200-d layer after the 3000-d layer and linked with fully connected network. The two approaches yield similar results (the network architectures and training codes are available at the Github repository). Similar to many cases in neural network training/applications, the choice of 200 parameters for this compressed layer is not unique. However, the encoding capacity will be affected if fewer parameters were used.

The decoding part is relatively simple. The 200-dimension vector is converted to a 4D tensor of size 8x8x8x32, then followed by two deconvolution layers, and finished with a convolution layer to obtain a 32x32x32 matrix, from which a submatrix of 31x31x31 was obtained as the reconstructed model. A preset threshold of 0.1 was used to convert the matrix to binary values.

The 60,000 models in the training dataset were fed to this auto-encoder with the loss function measured by cross entropy between the input models and the encode-decoded maps. The decoding part of the neural network can be applied to interpret any 200-d vectors to its corresponding 3D density map in the form of voxel object.

Model reconstruction from SAXS profiles

The overall workflow for model reconstruction using the auto-encoder method is as follows (see Figure S1). First, a number of latent parameter sets (or genes, in terms of genetic algorithm) are generated to initialize the genetic population to be optimized by genetic algorithm. The parameters for each gene is sampled based on the gene value distribution at the corresponding positions (see Figure 1d for representative probability distributions). After initialization, the iterative optimizations will be carried out. During each iteration, the genes are decoded to 3D voxel objects using the auto-encoder network trained using the molecular shapes abstracted from the PISA database. To ensure the continuity of the reconstructed models, only the largest connected domain of each decoded object will be kept as the 'cleaned model'. The SAXS profiles for the 'cleaned models' are then computed using the Zernike expansion method implemented in the SASTBX (elaborated below). Chi-scores between model profiles and target SAXS profiles are used to guide the genetic algorithm to evolve the genes until the chi-score is converged or a pre-set number of iterations is reached. The final models are saved in density maps (CCP4 format) or bead models (PDB format).

For a voxel object, the 3D Zernike representation has the advantage of de-coupling the model shape and size information; thus, the SAXS profiles can be quickly evaluated if the model size is updated while the shapes are not changed. The detailed derivation was elaborated elsewhere (Liu et al., 2012a, 2012b), and a brief summary is provided for clarity. A 3D object $\rho(\mathbf{r})$ after scaling to fit within a unit sphere can be expanded as the weighted summation of 3D Zernike functions, which are a set of orthonormal polynomials with orders (*n, l, m*) (Canterakis, 1999; Novotni and Klein, 2003). The expansion coefficient or the so-called Zernike moment C_{nlm} at order (*n, l, m*) is calculated with equation (1):

$$C_{nlm} = \int_{|\boldsymbol{r}|<1} \rho(\boldsymbol{r}) Z_{nlm}(\boldsymbol{r}) d\boldsymbol{r} \tag{1}$$

Subsequently, the 3D density distribution function $\rho(\mathbf{r})$ can be approximated using $\{C_{nlm}\}$ up to a maximum expansion order $n_{\max}$. It has been shown (see (Liu et al., 2012a)) that the SAXS intensity can be explicitly expressed as:

$$I(q) = \sum_{n} \sum_{n'} B_n(qr_{\max}) B_{n'}(qr_{\max}) H_{nn'} \tag{2}$$

where $B_n(qr_{\max}) = \frac{j_n(qr_{\max}) + j_{n+2}(qr_{\max})}{2n+3}$ describes the contribution from corresponding Zernike polynomials to SAXS profile with the Bessel functions of the first kind $j_n(x)$. In addition, the shape information is encoded in $H_{nn'}$, which is expressed using Zernike moments as follows:

$$H_{nn'} = \sum_l k_{nl} k_{n'l} \sum_m C_{nlm} C^*_{n'lm} \tag{3}$$

with $k_{nl} = (-1)^{\frac{n-l}{2}}$. According to equation (2), the radius of the object, r_{max}, is readily decoupled from the shape descriptors $\{H_{nn'}\}$. This allows us to optimize the radius as the 201st parameter (the other 200 parameters encode the 3D shape). The SAXS profiles can be calculated with other programs, as long as the program can be interfaced to the reconstruction software by providing the calculated intensities at desired scattering angles. In the case if the radius optimization is desired, the SAXS profile calculation is slightly more complicated, because the model scaling might be needed.

The genetic algorithm was used to optimize the parameters (200 parameters for given radius information or 201 parameters if the radius is a free parameter to be optimized) (Goldberg, 1989). The target function to be minimized is the standard chi-score:

$$\chi^2 = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{I_i^{\exp} - I_i^{\mathrm{mod}}}{\sigma_i} \right]^2 \tag{4}$$

Where $I_i^{\exp}$ and I_i^{mod} are the SAXS intensity profiles of the experimental measurement and the model calculation respectively, both at position q_i. The SAXS profiles were normalized to the range of [0,1] before the chi-score computation to remove the mismatched scaling factor and the offset level that caused by the residual background intensity. The $1/\sigma_i$ was used as the weighting factor. It is noted that the raw SAXS data were preprocessed using a sliding polynomial fitting (with order=2, window size=5) to smooth the profile. In order to balance the contributions from all data points to the chi-score, we compared three different weighting schemes, specifically, setting σ_i to 1.0, sqrt(I_i^{exp}), and I_i^{exp}. The results suggested that the final reconstructed models reached similar accuracy levels. In the program, the default weighting scheme is σ_i =1.0. It is noteworthy to point out that chi-score is just one of many possible metrics to quantify the difference between model profile and experimental data. Other formulations, such as likelihood functions based on probability theory can be also adapted easily by replacing the scoring function in the program. In this study, the testing results showed that the present implementation can achieve very good model quality.

The distribution of latent parameter values obtained during the auto-encoder training procedure was used to guide the initialization of model parameters, so that the genes (each with 200 or 201 values) for the first generation of the genetic algorithm were populated by sampling the latent parameter distributions to start the optimization procedure. In each generation, 300 genes are generated via simulated evolution procedures. The gene evolution was implemented using three operators:

selection, crossing, and mutation. The selection operator decides which genes are inherited from the previous generation by selecting the more fitted gene out of two randomly selected genes. The crossing operator is included to exchange gene segments obtained from the previous generation. The mutation operator changes parameter values at random positions by replacing the current value to a random value with a probability distribution that follows prior knowledge of empirical distributions at that gene position.

The model comparison was measured using the correlation coefficients (*cc*) after model alignment to the reference model (not used during model reconstruction process), which was performed using the fast rotation algorithm implemented in sastbx.superpose (Liu et al., 2012b). The correlation coefficient used to compare variables with binary values is referred to as Phi coefficient, similar to the pearson correlation coefficient for variables with continuous values (Guilford, 1936). In statistical analysis, the range for Phi coefficient is not strictly confined in [-1,1] (Davenport and El-Sanhurry, 1991). We computed the Phi_{max} between decodeSAXS reconstructed models and the reference models to provide a reference (the values are mostly above 0.80, see Figure S4 for a distribution). Here, the concept of Phi coefficient is borrowed to measure the similarity between reconstructed model and the reference model. The computed *cc* is used to quantify the overlapped volume from the two models being compared. The physical interpretation is the that *cc*=1.0 corresponds to perfect overlapping (aligned to itself), and *cc*=0.0 corresponds to zero overlapping between the two models. Both Phi_{max} and cc can be used to measure the model similarity. The cc values are used to quantify the model similarity through the analysis because of its clear physical interpretation, and the Phi_{max} is provided as a reference. For consistency analysis for models from multiple reconstructions, the hierarchical clustering algorithm was applied using correlation distance (defined as 1.0-*cc*). The figures were prepared using Chimera (Pettersen et al., 2004) or Pymol (Schrödinger, 2015).

Supplemental Reference

Davenport, E.C., and El-Sanhurry, N.A. (1991). Phi/Phimax: Review and Synthesis. Educ. Psychol. Meas. *51*, 821–828.
Guilford, J.P. (1936). Psychometric methods, (New York; London: McGraw-Hill Book Company, Inc.).
Novotni, M., and Klein, R. (2003). 3D zernike descriptors for content based shape retrieval. In Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, (New York, NY, USA: ACM), pp. 216–225.
Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., and Ferrin, T.E. (2004). UCSF Chimera - A visualization system for exploratory research and analysis. J. Comput. Chem. *25*, 1605–1612.
Schrödinger, L. (2015). The {PyMOL} Molecular Graphics System, Version~2.2.